

# SIMPLIFIED INPUT FOR CERTAIN AERODYNAMIC NOSE CONFIGURATIONS TO THE GRUMMAN QUICK-GEOMETRY SYSTEM (A KWIKNOSE USER'S MANUAL)

Frederick L. Shope
ARO, Inc., a Sverdrup Corporation Company

VON KARMÁN GAS DYNAMICS FACILITY
ARNOLD ENGINEERING DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
ARNOLD AIR FORCE STATION, TENNESSEE 37389

February 1978

Final Report for Period January to September 1977

Approved for public release; distribution unlimited.

Prepared for

ARNOLD ENGINEERING DEVELOPMENT CENTER/DOTR ARNOLD AIR FORCE STATION, TENNESSEE 37389

#### NOTICES

When U. S. Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, or in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Qualified users may obtain copies of this report from the Defense Documentation Center.

References to named commerical products in this report are not to be considered in any sense as an indorsement of the product by the United States Air Force or the Government.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public. including foreign nations.

#### APPROVAL STATEMENT

This report has been reviewed and approved.

Project Manager, Research Division

Directorate of Test Engineering

Approved for publication:

FOR THE COMMANDER

MARION L. LASTER

Director of Test Engineering

Deputy for Operations

# **UNCLASSIFIED**

REPORT DOCUMENTAT	READ INSTRUCTIONS BEFORE COMPLETING FORM	
AEDC-TR-77-89	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
SIMPLIFIED INPUT FOR CERTA NOSE CONFIGURATIONS TO THE	5. TYPE OF REPORT & PERIOD COVERE Final Report - January to September 1977	
GEOMETRY SYSTEM (A KWIKNOS: MANUAL)	6. PERFORMING ORG, REPORT NUMBER	
7. AUTHOR(e)		B. CONTRACT OR GRANT NUMBER(a)
Frederick L. Shope, ARO, I	nc.	
Arnold Engineering Development Force Systems Command Arnold Air Force Station,	ment Center/DOT	10. PROGRAM ELEMENT PROJECT, TASK AREA & WORK UNIT NUMBERS Program Element 65807F
1. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE
Arnold Engineering Develop	February 1978	
Arnold Air Force Station,		13. NUMBER OF PAGES 93
MONITORING AGENCY NAME & ADDRESS(II di	lerent from Controlling Office)	15. SECURITY CLASS. (of this report)
		UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the ebstract entered in Black 20, if different from Report)

18. SUPPLEMENTARY NOTES

Available in DDC

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

handbook user needs computer programs

geometric forms

input

mathematical model

Fortran

flow fields

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report is a user's manual for a Fortran computer program KWIKNOSE which, for certain axisymmetric and nonaxisymmetric nose configurations, provides simplified geometric input to the Grumman QUICK-geometry system, which in turn provides geometric information to various numerical flow codes. For a wide variety in choice of input parameters, KWIKNOSE sets up the QUICK input for an arbitrary sequence of conical and ogival sections. In this process, KWIKNOSE

aerodynamic configurations

# **UNCLASSIFIED**

## 20. ABSTRACT (Continued)

performs the tedious computations necessary to locate the intersection points of successive arcs and to insert optional fillets or rounds over nontangent intersections. In addition, the code is capable of inserting arbitrary multiple slicing planes into the top, bottom, and side of the vehicle. Slicing plane intersections may be filleted or rounded. Thus, for a minimum of input and manual calculation by the user, KWIKNOSE is tailored to modeling the geometry of a sliced multiconic vehicle capped with an asymmetrically ablated nose. This manual provides check cases for the various geometry options, a description of input and output, and a listing of the source deck.

APSC Amold APS Tenn

## **PREFACE**

The work reported herein was conducted by the Arnold Engineering Development Center (AEDC), Air Force Systems Command (AFSC), under program element 65807F. The results were obtained by ARO, Inc., AEDC Division (a Sverdrup Corporation Company), operating contractor for the AEDC, AFSC, Arnold Air Force Station, Tennessee. The research was done under ARO Project No. V33A-A8A. Elton R. Thompson was the Air Force project manager. The manuscript was submitted for publication on September 2, 1977.

## **CONTENTS**

		Page
1.0	INTRODUCTION	5
2.0	PROGRAMMED GEOMETRIES	8
3.0	PROGRAM DESCRIPTION	
	3.1 Input	11
	3.2 Output	16
	3.3 Processing Information	17
	3.4 Programming Considerations	18
4.0	SUMMARY	21
	REFERENCES	21
	ILLUSTRATIONS	
Figu		0.0
	Typical Geometries Available via KWIKNOSE	23
2.	Relative Regions of Applicability of Programs	0.4
	KWIKNOSE and PREKWIK	24
	Oblique View of Asymmetrically Ablated Nose	25
	Oblique View of Asymmetrical Nose from Sample Case	26
5.	Oblique View of Asymmetrically Ablated, Bent, and	
	Sliced Cone (Sample Case)	27
6.	Nose-Afterbody Sample Case with Notation for	
	Input Variables	28
7.	Body Line Notation	29
8.	Cross Section and Body Line Models	30
	TABLES	
1.	Input to KWIKNOSE via Namelist/INPUT/	31
2.	. KWIKNOSE Input Options for Three Main Body Lines	
	(ZUCL, ZLCL, YSID)	34

## AEDC-TR-77-89

TABL	ES (continued)	Page
3.	<pre>Input Description for Variable DIR(I)</pre>	35
4.	Nomenclature for Input Variable DIR(I)	36
5.	Jobstream for Sample Case	37
6.	KWIKNOSE Printout for Sample Case	38
7.	Definitions of Logical Names for QUICK Cross-Section	
	and Body Line Models	47
8.	KWIKNOSE Source Listing	48
9.	Subroutine Description	85
10.	Variable Description	87

## 1.0 INTRODUCTION

This document describes a computer program which allows greatly simplified input, for certain common axisymmetric and nonaxisymmetric nose-vehicle geometries of general interest, to the more general Grumman QUICK-geometry system. QUICK, a Fortran computer program developed by Vachris and Yaeger (Refs. 1 and 2), provides the user with a simple and well organized method of constructing and interrogating a mathematical model of an arbitrary, three-dimensional flight vehicle. This geometry system has primary application to the numerical computation of the flow fields over arbitrary aerodynamic configurations. QUICK was developed in two phases, the QUICKDEF phase, which provides facilities to build and check the geometry model, and the QUICKLOK phase, which provides surface coordinates and derivatives to user programs without requiring of the user a detailed knowledge of how the model was constructed. Current applications of the QUICKLOK phase include the time-dependent blunt body program BLUNT (Ref. 3); the supersonic, three-dimensional, external, inviscid flow-field code STEIN (Refs. 3 and 4); and the multishocked, three-dimensional, supersonic finite difference method of Kutler, Reinhardt, and Warming (Ref. 5). The QUICKDEF phase, of primary concern here, is a highly user-oriented program which greatly simplifies the geometric input by providing general curve fitting services to define arbitrary bodies from a minimum of logical and numerical input. In essence, QUICK requires the user to define each unique cross section of the vehicle in a plane normal to the coordinate axis of the vehicle by merely identifying the types of curves (lines, ellipses, parabolas, ...) which form the segmented cross section. The cross sections then become numerically specific when the user further defines, logically and numerically, certain key control points which determine the intersections and slopes of the curves comprising the cross sections. The only two limitations imposed by QUICK upon the vehicle geometry are (1) that the vehicle must have a vertical plane of symmetry and (2) that each cross section must be definable in terms of a single-valued radius

vector swung about a single point within the cross section. While these limitations prevent complete treatment of certain vehicle geometries (such as engine inlets, the oblique wing supersonic transport, or the double vertical stabilizer of the F-15), QUICK is an important advance in the description of vehicle geometry and makes practical the solution of flows over such complicated bodies as the space shuttle orbiter.

In contrast, QUICK is also well suited to the treatment of relatively simple bodies such as sphere-cones, biconics, and slab deltas, which are of basic interest to aerodynamicists. It was therefore deemed logical to develop yet another sequential computer program to set up input to QUICK for certain such simple geometries, thus reducing user-prepared input to a matter of two or three cards for the programmed geometries. This fourth program PREKWIK is documented in a previous user's manual.

During the development of PREKWIK, the need was recognized to study more complicated nose shapes than the simple spherical cap treated by Interest centered around the effects of ablated nose shapes (Fig. 1) upon the aerodynamics of otherwise simple bodies such as cones and biconics with slices on the conic portions of the vehicle. A feature which most of the symmetrical noses have in common is that they can all be modeled (with varying degrees of accuracy) as a stack of conical and ogival sections which intersect in different fashions. Thus, it appeared practical to program the logic needed to model these geometries and set up the input to QUICK. Initially, the nose code was envisioned as a modification to PREKWIK, but the need for a fundamentally different programming approach soon forced the nose geometry problem into a separate computer program. The essential goal of PREKWIK was to receive a minimum of geometric input data via very few input options for a given geometry, perform a few relatively trivial intersection computations, and then generate the card image input required by QUICK. programmed geometries, PREKWIK eliminated the tedious chore of preparing

a new QUICK deck for every minor quantitative change in geometry (such as a different bluntness, cone half-angle, or vehicle length). contrast, the nose geometry problem was perceived as logically more complicated. Not only would the nose program have to compute the intersections of more complicated functions, but the capability to round or fillet sharp corners was deemed essential. However, the analytic geometry of rounds and fillets - determing the equation and intersection points of a circle tangent to two intersecting curves - can result in very high degree polynomials along with the attendant logic problems of root sorting to choose the correct intersection. The computational complexity of the intersection problem coupled with the fairly large number of ways a given geometry can be specified via various combinations of known parameters forces the geometry program to accept many different input combinations from which must be computed the same basic set of QUICK input coordinates and slopes. If the nose code did not accept a fairly broad range of input combinations, then the typical user (who didn't have quite the right input combination) might have to spend a great deal of time deriving and debugging the intersection equations. After he had done all this, the computation and assembly of the final QUICK input would seem a fairly minor task. In the meantime, he would have surely forgotten all about the nose code.

Such is the logic upon which is based the structure of the nose geometry code KWIKNOSE, the subject of this user's manual. KWIKNOSE should be viewed as a complement to PREKWIK: whereas the latter is intended to model the overall vehicle, KWIKNOSE is specialized for arbitrary asymmetrical nose shapes. With these two programs, then, a flow program deriving its geometric data from QUICK could handle, as an example, the symmetrically ablated bent cone of Fig. 2. A KWIKNOSE/QUICK/BLUNT/STEIN chain could carry the computation to some point on the forecone, and a PREKWIK/QUICK/STEIN chain could proceed from that point through the bend to the end of the aft cone. (The user may have to construct a fictitious sphere-cone forebody on the PREKWIK model to

place the aft body at the same position relative to the geometric orgin of the KWIKNOSE model.)

The asymmetrical nose geometries which KWIKNOSE can generate are limited to cases where the top, bottom, and side body lines can be. described in terms of sequential series of straight line and circle segments (see Figs. 3 and 4) as was done for the symmetrical case. upper and lower half of every cross section must be describable in terms of an ellipse, though the ellipses at successive cross sections need not be concentric or geometrically similar (may have different centers and different major and minor axes). All filleting, rounding, and slicing. capabilities are available in the asymmetrical case. However, if the tip of the asymmetric nose does not fall on the centerline of the afterbody (or the first conical section of a stack), the resulting geometry model will be incompatible with the PREKWIK geometry model for the afterbody. However, KWIKNOSE does not distinguish between nose and afterbody sections, and a fairly general asymmetric nose atop a stack of cones and ogives (with bends) can be moldeled entirely with KWIKNOSE without reference to PREKWIK (Fig. 5). Some additional geometric calculation may have to be done by the user.

The remainder of this report describes in greater detail the programmed geometries and the code itself. A description of the input and output is given, and a sample case is presented to illustrate use of the code. The necessary processing information is listed, and certain portions of the programming logic are elaborated upon.

## 2.0 PROGRAMMED GEOMETRIES

The configurations which can be modeled via KWIKNOSE fall into two general categories: axisymmetric and nonaxisymmetric. The symmetric configuration is an arbitrary sequence of conical and ogival sections which may or may not be tangent at the intersection points. The ogival

sections may be convex or concave relative to the centerline of the vehicle. Nontangent intersections may, at the user's option, be smoothed with circular fillets or rounds. The fillets (or rounds) may be inserted between cone-cone intersections, cone-ogive intersections, or ogive-ogive intersections. The fillets need not be very small in relation to the intersecting curves and may therefore form major segments of the body. The only restriction is that the adjacent segments being filleted intersect in the real plane. For some cone-ogive and ogive-ogive combinations, there may exist a maximum filleting radius beyond which a real solution does not exist. In general, no segment may contain a point with infinite slope, the nosetip being the one exception. The tip of the nose may be either sharp or blunt and is always used as the geometric origin for the entire geometry. No segment may wrap back on itself so as to require a double-valued radius from any single point on the centerline axis. The program dimensions limit the geometry model to ten or fewer segments, and each fillet occupies one segment (see Table 1, note 8). Since KWIKNOSE does not distinguish between nose and afterbody portions of the vehicle, the user may allocate the ten segments between these two portions as needed.

The program contains logic to insert slicing planes into the basic axisymmetric vehicle described above. Provisions are made for an upper slice, a side slice, and a lower slice, each being composed of nine or fewer straight-line segments. In the side view the top and bottom slicing planes are seen as segmented lines, and the side slicing plane is seen as a segmented line in the top view. Thus, there are no provisions for skewed slices. The slices may not cross the centerline of the vehicle and may not intersect each other inside the region of the unsliced vehicle at a particular station. Corners at the intersection of adjacent segments may be rounded or filleted at the user's option. Although the slices are assumed to be planar, the fillets need not be small and may therefore comprise major portions of the slicing segments. It should be understood that the slicing body lines are input to QUICK

independently of the other body lines and thus do not have to intersect the unsliced vehicle at all points or any points. QUICK automatically checks whether the slicing body lines intersect the unsliced body and chooses the appropriate curve for the outer skin of the vehicle. The slices therefore may enter and leave the body as needed. There are no provisions for filleting or rounding in the circumferential direction (as viewed in planes normal to the centerline of the vehicle).

The asymmetrical case retains all combinations of cone and ogive intersection options (or line-circle intersections in a particular radial plane), the rounding and filleting of the outer body lines, and slicing with rounds and fillets. The asymmetrical vehicles treatable by KWIKNOSE must be such that the top, bottom, and side surface body lines may be represented by segments of lines and circles. When viewed in cross section, the vehicle must be describable with two ellipses (not necessarily identical) for the upper and lower half-planes. At a given cross section, the horizontal line dividing the two half-planes need not intersect the centerline (or geometric axis through the nosetip), and provisions are available for defining this deviation. For the asymmetrical case the top and bottom surface body lines may cross the geometric axis, and the map axis (from which QUICK swings the radius vector which must remain single valued) may also depart from the geometric axis if needed. Thus, the asymmetrical option allows, for example, the modeling of an ablated nose atop a bent cone.

For certain asymmetric configurations there is a potential inaccuracy of which the user should be aware. Consider a geometry such as is illustrated in Fig. 3 but without the round between the fore and aft sections. This geometric model was constructed with the geometric axis (X-axis) through the nosetip and parallel to (but not coincident with) the aft cone centerline. In the actual hardware (before rounding), the intersection of the fore and aft sections when viewed from the side would appear as a slanted line. Discontinuities in the surface slope

would occur at this line in moving along a body line. In the math model, however, the slope discontinuity would occur only on the upper, lower, and side body lines and would be smeared everywhere in between. This smearing results from two conditions: (1) QUICK assembles cross sections in planes normal to the geometric axis; (2) as modeled in Fig. 3, the slanted line between the fore and aft bodies is oblique to the geometric axis. While a carefully constructed manual model for input to QUICK probably could represent this case accurately, programming of the general case for arbitrary intersections of cones and ogives was considered beyond the scope of KWIKNOSE.

The user has one alternative if he requires an accurate model from KWIKNOSE: provide the input data relative to a geometric axis normal to the slanted line. Note, however, that the geometric axis might penetrate the skin and require bending of the map axis to keep QUICK's radius vector single-valued. In addition, once the geometric axis has been made normal to a particular intersection plane, no other similar intersection with a different slant angle can be modeled with equal accuracy. For two conical sections of different half-angles and oblique centerlines, the intersection "plane" is really curved. Thus, reorienting the geometric axis can only reduce (but not eliminate) the inaccuracy in modeling.

#### 3.0 PROGRAM DESCRIPTION

### 3.1 INPUT

The input to KWIKNOSE is described in Tables 1 through 4, and a listing of the KWIKNOSE/QUICK jobstream with the input for the sample case of Figs. 4 through 6 is given in Table 5. It is suggested that the user, prior to assembling the input deck, make a qualitatively accurate sketch of the upper, lower, and side body lines and indicate the known geometric data. This task has been found useful in choosing appropriate

input options and deciding on the values of the logical input variables (DIR and SLOPE), which are needed by the code for selection of a segment intersection from multiple roots.

The first input card is for identification and may contain any 80 characters, though QUICK will carry only 60 of them. The remaining input is via namelist /INPUT/. For a symmetrical body (with or without asymmetrical slices), only one namelist input is required. asymmetrical body three separate namelists must follow the single identification card. The first must define the upper body line (ZUCL, see Figs. 7 and 8), the second the lower body line (ZLCL), and the last the side body line (YSID). Note that the input variable Z is used for the radial coordinate regardless of whether the input data represents the Y or Z direction. The slicing body lines (ZUSL, ZLSL, and YSSL) and the Z coordinates of the side and map axis (ZSID and ZMAPAX) may be included in any of the three namelists, if they are input at all. The upper. lower, and side body lines are processed independently of each other and need have no particular correspondence with each other except for the same final X coordinate at the end of the last segment. Thus, each may be an entirely different combination of lines, circles, and fillets. The sign notation for input coordinates and angles is that of a cylindrical coordinate system. Angles (TH(I)) are positive away from the centerline and negative toward it relative to a line parallel to the centerline.

The input rules for the three main body lines are somewhat different than for the slicing body lines and will be discussed separately. For each of the three main body lines (upper, lower, side) to be defined, the user must input the element of the character array CURVE(I) as 'LINE' or 'OGIV' for each segment. Based on this choice for each individual segment, the user then chooses an input option from Table 2 for each of the segments, preferably an option which matches the combination of geometric data he has for the segment. If any redundant data is to

be input, case 17 or 18 (see ICASE values in Table 2) must be chosen. If the redundant data is not self consistent to within ERRMAX, execution will be halted. All data items indicated by X's in Table 2 for a particular ICASE value must be input or execution will be halted. Note that ICASE(I) is not an input variable. The data elements described in Table 1 are then coded according to the standard rules for namelist input as found in any Fortran manual. The user should note that X(I), Z(I), and TH(I) are the coordinates and slope at the end of the Ith segment (CURVE(I)). Further, certain input cases may not be used following certain other cases. For example, cases 5 and 9 constitute inputting only the equation of the segment and in themselves do not include sufficient information to locate the intersection of the end of the segment. Thus, those cases which assume that the coordinates of the end of the previous segment are known may not follow cases 5 and 9. Such limitations, the need for which will be clarified in Section 3.4, are given in the footnotes to Tables 1 and 2.

The direction indicator variable DIR(I) requires additional clarification. Whenever ogival segments are involved, the possibility of multiple intersections with adjacent segments exists. In addition, when an ogive is to be inserted between two given points, there are usually two circle equations which will satisfy all the numeric data. The user therefore must indicate to KWIKNOSE which of the possibilities is to be chosen using DIR(I). Table 2 indicates which input cases require DIR(I) and Table 3 defines the acceptable values of DIR(I) for each ICASE value. The nomenclature for these acceptable values is defined in Table 4. The 'OPUP', 'OPDN', 'TOPS', and 'BOTS' values apply to all three main body lines (upper, lower, and side) as though they were rotated about the centerline for viewing as oriented in Table 4. That is, DIR(I) = 'OPUP' indicates that the centerline sees a convex segment and 'OPDN' a concave segment. Similarly, 'TOPS' always indicates the intersection farther from the centerline than the 'BOTS' intersection.

If a nontangent intersection is to be filleted or rounded, this fact is indicated by giving the radius of the round or fillet RHO(I) to be inserted at the end of the Ith segment. Fillets/rounds are not included when numbering segments for input, and no data other than the radius are input for the fillet. The intersections and center are computed internally by KWIKNOSE. There are several restrictions on the filleting (or rounding) process. Segments already tangent may not be filleted. Only adjacent segments which actually intersect may be filleted because the location of this intersection is used to choose between multiple roots. Further, some types of intersections may have maximum filleting radii beyond which real solutions do not exist. If KWIKNOSE detects any of these conditions, usually through negative square roots or division by zero, execution will be halted.

For asymmetrical bodies, such as a bent cone, the user may opt to define the Z-coordinate of the side (ZSID, Figs. 6 and 8) as different from zero, in which case he must input two or more values of XSID and ZSID as described in Table 1. XSID(1) and ZSID(1) must both be zero; and if ZSID is to be input at all, at least two values of each (XSID(1) and XSID(2), ZSID(1) and ZSID(2) must be given. If the vehicle is bent so far that the geometric axis penetrates the skin, the user must bend the map axis via XMAPAX and ZMAPAX so that a radius vector swung about the map axis at any station along the vehicle will be single-valued. Care should be taken in using models generated by QUICK in which the map axis is bent. The stock version of the QUICKLOK phase of QUICK returns to the user program radius vectors relative to the map axis rather than the geometric axis. Thus, a bent cone whose map axis follows the centerlines of the cones may appear to the user program as not being bent at all. Input follows the same rules as for XSID and ZSID above. Note that both ZSID and ZMAPAX may be negative. These variables are used only if TYPE = 'ASYM', for which case ZSID defaults to zero if not input, and ZMAPAX defaults to ZSID if not input. These input variables may be included in any one of the three namelists.

Input for the slicing planes as well as the processing of the data by KWIKNOSE is completely independent of the main body lines. Any combination of the three slices may be used regardless of the symmetry aside from the slices (the value of TYPE has nothing to do with symmetry of the slices). The minimum slicing plane input, if present at all, is the coordinates of the beginning and end of one segment. At the user's option, different combinations of X and Z (or Y) coordinates and slope angle may be input. Note that TUSL(I) is the slope angle of the segment ending at XUSL(I), ZUSL(I), for I > 1. Thus, TUSL(1) = TUSL(2), and similarly for the other two slices. Redundant data will be checked for consistency; but if any one of the three is not input for a given segment, it will be computed. The slicing planes need not cover the entire length of the vehicle, but the first and last segments will be extrapolated to the extremities if not. Nontangent intersections of adjacent segments may be filleted or rounded by inputting the fillet/round radius as indicated in Table 1. Note that RHOUS(I) fillets the intersection at XUSL(I), which is the end of the I-1st segment; otherwise, all the rules of body line filleting/rounding apply.

The remaining input variables are program control integers. IPRINT set different than zero will cause KWIKNOSE to add cards to the QUICK input deck to exercise the math model and generate printouts of coordinate and slope data at various points along each body line segment. IPRINT=0 (the default) suppresses such printout. The variable IR is the unit number (Data Set Reference Number) of the facility card reader, and IW is the unit number of the printer. IF is the unit number of the KWIKNOSE output file of card images to be passed to and read by the QUICKDEF phase of QUICK. IP is the unit number for the facility card punch. If IF and IP are different, the data from unit IR is copied as card images with an 80-character array to IF, from which the input cards are then printed on IW. IF is then rewound and read as namelist data.

## 3.2 OUTPUT

Table 6 is the KWIKNOSE printout for the sample case. The printout is composed of three sections: the input data, the results of the computations, and the QUICK input file. The first section, labeled "INPUT CARDS" is a listing of the data cards exactly as punched. The data between the first two lines of periods is the input data after the first namelist has been read. Notice that geometric variables not input through the namelist default to 1.E70. This defaulting is critical to the program's logic and should not be trampered with by inputting userchosen default values which have no geometric meaning. The printout from this point to the third line of periods contains the results of the geometric computations. First, the number of segments exclusive of rounds and fillets is printed followed by the code's determination as to the input options the user has chosen for each segment. Next, the results of the computation of segment intersections and equations are printed. In the printout labeled "SEGMENT END POINT COORDINATES AND SLOPES", X and Z are the coordinates at the beginning of the segment, except for the last point (with values less than 1.E70), which is the end of the last segment. TA and TF are the slope angles in degrees at the aft and fore ends, respectively, of the segment. The segment equations in the next section of the printout are defined in terms of the center coordinates and radius for ogival sections, and slope angle and Z-intercept for straight-line segments. The X value printed here is the coordinate of the end of the segment. In these terms the line and ogive equations are

line: 
$$Z = X \tan T + B$$
  
ogive:  $(X-H)^2 + (Z-K)^2 = R^2$ 

Next, the results of the fillet and round computation is printed. The equation of the fillet is defined as

fillet: 
$$(X-KSI)^2 + (Z-ETA)^2 = RHO^2$$

and straddles the printed X value. X1, Z1, T1, and X2, Z2, T2 are the coordinates and slope angle (degrees) at the fore and aft intersections, respectively, that is, the points of tangency with the intersected segments. The fillet/round intersections are then printed as additional segments included with the previous unfilleted body line under the heading "FINAL QUICK ARRAY". Note that the original intersection point has been removed from the arrays. This completes the geometric computation for the upper body line (ZUCL). When TYPE = 'ASYM' in the namelist input, this computation and printout are repeated first for the lower body line (ZLCL) and then for the side (YSID). The third printout includes the results of the geometrical calculation for slices. section of the printout headed "SLICE GEOMETRY" gives the slope angle for each segment ending at the indicated coordinate values. results are exclusive of rounds or fillets, for which immediately follows the computational results. BI and BJ are the Z-intercepts of the two lines formed by the locus of the fillet/round center while rolling on the two intersecting segments. A3 and A4 are the Z and X coordinates of the fillet/round center; and X1, Z1, T1, etc., are the tangency points as described for the main body line fillet/round printout. The final portion of the computational printout is the slicing body lines with the rounds and fillets inserted. The KWIKNOSE printout concludes with a listing of the card images of the QUICK input file. This listing will not be generated if IF = IP. The contents of this output deck may be understood from Table 7 and Refs. 1 through 3 and will not be discussed here. The user should understand that this deck comprises the input to the QUICKDEF phase of QUICK and not the QUICKLOK phase, which is the portion of QUICK employed in the fluid dynamics programs to obtain geometric information.

#### 3.3 PROCESSING INFORMATION

KWIKNOSE is a Fortran IV program containing about 2,000 cards (Table 8). The code was developed on an IBM 370/165 computer system

under the H Level 21.7 compiler. Compilation of the entire deck requires 230 K bytes of core and 40 sec of central processing unit (CPU) time. Execution of the code requires 80 K bytes of core, and all geometries processed to date have required less than 2 sec of CPU time. Users should note that because of the character manipulation performed in the program, it cannot be compiled under the G level 21 compiler. Execution requires one peripheral storage device for unit IF (defaulted to unit 1). This should be a sequential file formatted for 80-character records as illustrated in Table 5.

# 3.4 PROGRAMMING CONSIDERATIONS

The overall flow of logic in KWIKNOSE is basically simple and can be best understood via the series of CALL statements in the main program (Table 8) and the subroutine description of Table 9. Execution begins with the initialization of certain data (subroutine INIT), after which the first namelist is read (INPT). Next, the input option chosen by the user is determined for each segment (CASE) in terms of the value of ICASE(I). This variable is then used to control branching in determining the segment equations (SEGE) and locating the segment intersections (INTR). This completed, the results of the computation to this point are copied into the QUICK arrays XQ, ZQ, and TQ (QARY) and printed (PRNT). Next, the round/fillet computations are performed and added to the QUICK arrays (ROFL). If slices are present (indicated by ISL  $\neq$  0), the necessary computations are performed and the results assembled in the QUICK slicing array V (SLCE). If rounds or fillets are present on the slices, these computations are performed and the results inserted into the original arrays (RFSL). If the case is symmetrical, the QUICK input card image file is generated (QUCK) and printed (LIST). case is asymmetrical, the completed body line with rounds and fillets is copied into the three-dimensional QUICK array VB (COPY). When all three main body lines are complete, the slicing with optional rounds/tillets is performed and copied into the three-dimensional arrays. Next, ZSID and

ZMAPAX are defaulted if not input, and the remaining body lines are checked for validity (BLCK). Finally, the QUICK input file for the asymmetrical case is generated (QK3D) and listed (LIST), thus completing the processing.

For a more detailed examination of the code, the variable description in Table 10 may be useful. Users interested in the intersection computation should find the logic as coded in SEGE and INTR fairly straightforward, but a few clarifying comments about the rounding and filleting logic are necessary. The straightforward approach to locating the center and tangency points of a circle filleting two other circles is to use analytic geometry, by writing four circle equations and two tangency equations in terms of six unknowns. However, this is equivalent to a polynomial of degree 48 for one unknown, which is an absurd thing to try to solve. The approach used in ROFL is to write the equation of the locus of fillet centers as the fillet rolls on the two adjacent segments, where the intersection of these two loci of centers is the center of the filleting circle. This approach reduces the problem to second degree but requires additional logic to decide when to add or subtract the fillet radius to or from the segment radius (ogive) or Zintercept (line). Fortunately, the information necessary to make this decision can be extracted from the slopes of the adjacent segments at the intersection point. For the line-ogive intersection, the final second degree problem can then be solved by choosing the set of tangency points which straddle the intersection point. For the ogive-ogive intersection, the filleting circle is chosen whose center is closest to the original intersection point. The advantages of using the loci-ofcenters approach are clear; but the major disadvantage is that segments which do not intersect cannot be filleted, whereas the general 48th degree problem includes this case.

Because of the complexity of the above logic as coded and to the many input options available, KWIKNOSE contains many small blocks of

coding to check on the validity of the input and the running computation. When a violation of the rules is detected, a condition code is generated with the Fortran statement STOP xxxx, where xxxx is the four-digit condition code. Execution is halted and an error message is usually generated. A program-generated condition code without an accompanying error message would indicate either a programming error or computer error (unlikely). The condition code is usually the same as the statement number of the format that generated the accompanying message. For five-digit statement numbers, the left most digit is truncated to obtain the condition code.

The accuracy of the code's geometry computation has been checked by comparison with manual computation on an electronic calculator. To the best of the author's knowledge, each block of coding has been checked against a tailored check case. However, the user is encouraged to carefully scrutinize the final QUICK math model for accuracy. The author has, to date, constructed about 30 manual card input decks to QUICK and hundreds of PREKWIK and KWIKNOSE decks, and no errors have been traced to QUICK. Thus, errors in the final math model will most likely be caused by either problems in the setup codes or their input decks. Checking of the QUICK math model can be effectively accomplished if a plotting device is available with stock software. With very little effort, the author has constructed a cathode ray tube (CRT) plotting program using the QUICKLOK portion of QUICK and has found it indispensable in debugging the geometry models. This approach is highly recommended to all users.

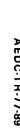
### 4.0 SUMMARY

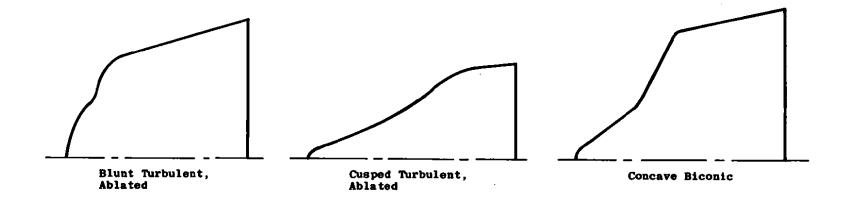
A user's manual has been presented for a Fortran computer program KWIKNOSE which, for certain axisymmetric and asymmetric nose configurations, provides simplified geometric input to the Grumman QUICK-geometry code, which in turn provides a geometric math model to various numerical flow-field programs. A discussion was presented as to how KWIKNOSE may easily be used to model any vehicle whose upper, lower, and side body lines can be described in terms of sequential segments of circles and straight lines and whose upper and lower half cross sections can be modeled with ellipses. It was further shown how the body may be sliced on three sides. A description was given of the input and output, and a sample case was presented to illustrate use of the code.

## **REFERENCES**

- Vachris, A. and Yaeger, L. "QUICK-GEOMETRY User's Manual." Grumman Aerospace/Aerodynamic Section Technical Data Report No. 393-74-1, 1974.
- Vachris, A. F. and Yaeger, L. S. "QUICK-GEOMETRY: A Rapid Response Method for Mathematically Modeling Configuration Geometry." NASA SP-390, 1975, pp. 49-73.
- Marconi, F. and Yaeger, L. "Development of a Computer Code for Calculating the Steady Super/Hypersonic Inviscid Flow Around Real Configurations, Volume II - Code Description." NASA CR-2676, May 1976.
- 4. Marconi, F., Salas, M., and Yaeger, L. "Development of a Computer Code for Calculating the Steady Super/Hypersonic Inviscid Flow Around Real Configurations, Volume I - Computational Techniques." NASA CR-2675, April 1976.

5. Kutler, P., Reinhardt, W. A., and Warming, R. F. "Numerical Computation of Multishocked, Three-Dimensional, Supersonic Flow Fields with Real Gas Effects." AIAA Paper No. 72-702, June 1972.





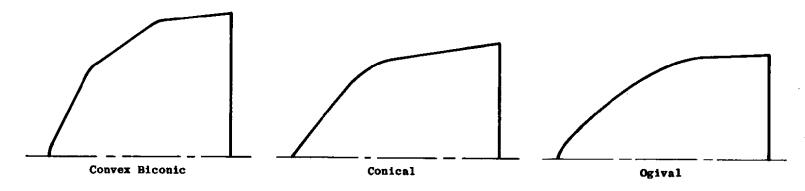


Figure 1. Typical geometries available via KWIKNOSE.

Figure 2. Relative regions of applicability of programs KWIKNOSE and PREKWIK.

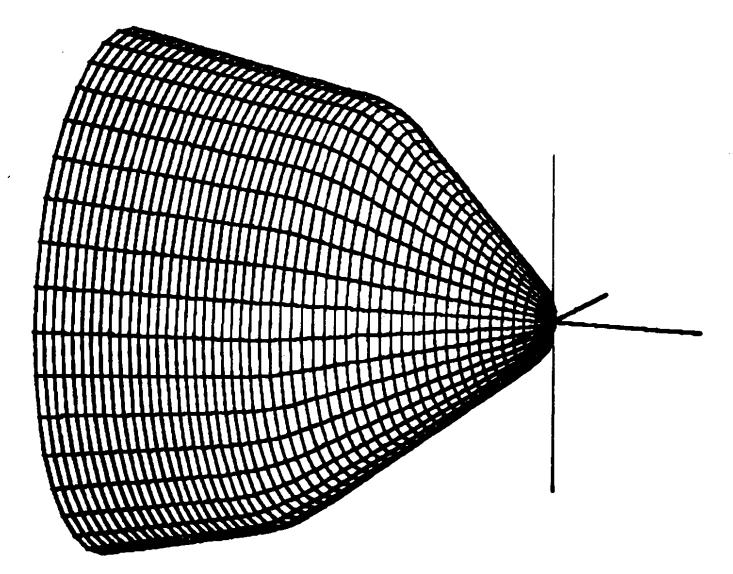


Figure 3. Oblique view of asymmetrically ablated nose.

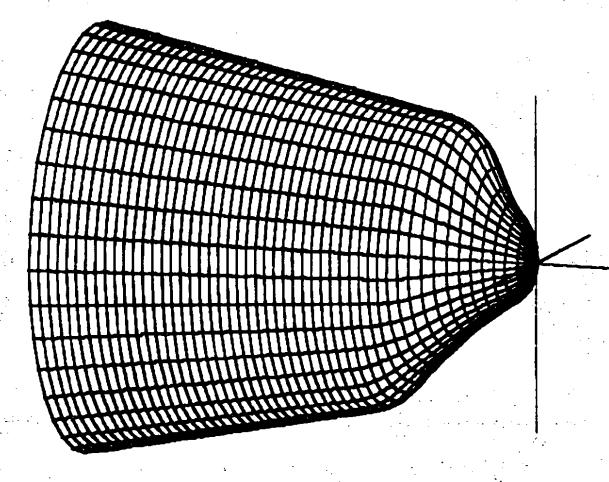


Figure 4. Oblique view of asymmetrical nose from sample case.

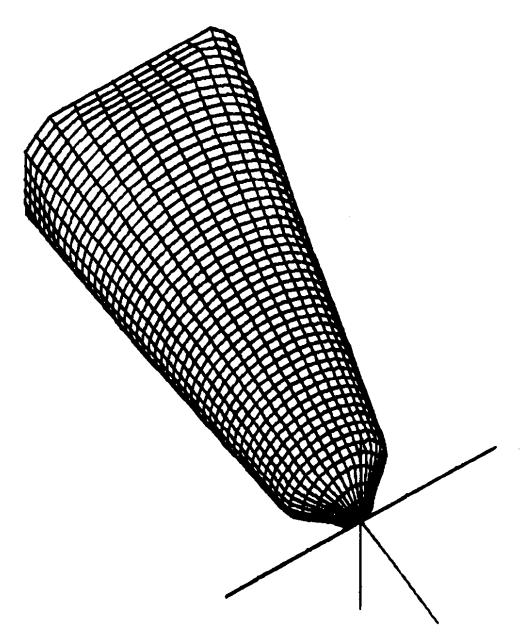


Figure 5. Oblique view of asymmetrically ablated, bent, and sliced cone (sample case).

Figure 6. Nose-afterbody sample case with notation for input variables.

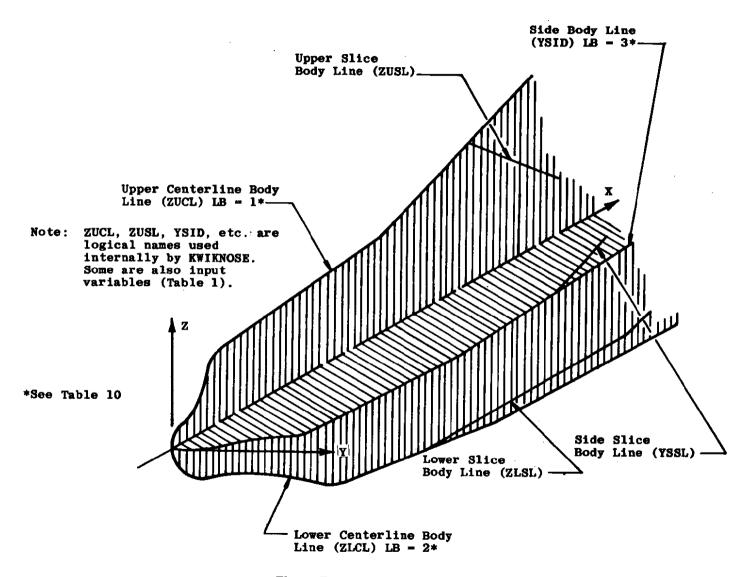


Figure 7. Body line notation.

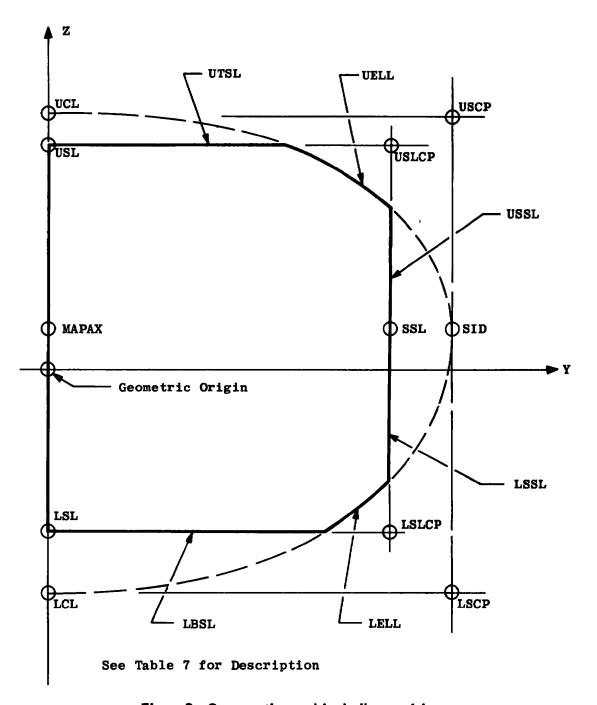


Figure 8. Cross section and body line models.

AEDC-1R-77-89

Table 1. Input to KWIKNOSE via Namelist/INPUT/

Namelist Variable	Value	Default Value	Meaning	TYPE <sup>7</sup>
X(I)	(0,∞)	1 . E70	Centerline distance to end of Ith ⇔egment (I≤10)	R+4
Z(I)	[O, TO]	1.K70	Radial distance (normal to centerline) to end of Ith segment	R+4
TH(1) <sup>2</sup>	(-90,90)	1.E70	Slope angle (deg) at end of Ith segment, positive away from centerline	R+4
CURVE(I)	'LINE'	'bbbb' <sup>1</sup>	Segment I is a straight line	R+4
	'¢GIV'		Segment I is an ogive	
R(I) <sup>3</sup>	(0,∞)	1.E70	Radius of ogive of Ith segment (CURVK(I)-'GIV')	R+4
H(1) <sup>3</sup>	(_∞, ∞)	1.E70	X coordinate of center of ogive (CURVE(I)-'\$GIV')	R+4
K(I) <sup>3</sup>	(_∞, ∞)	1.E70	Z coordinate of center of ogive (CURVE(I)='\psi GIV')	R+4
B(I) <sup>2</sup>	(-10, 10)	1.E70	Z intercept of line segment (CURVE-'LINE')	R+4
DIR(I)4	See Table 3	' PDN'	Direction indicator for CURVE(I)-'GIV' when aft intersection with adjacent segment may have two solutions	R+4
SLøPE(I)	'T'	'F'	Indicates ogive is to be made tangent to aft end of previous segment (do not uso for segment I=1)	L+1
ERRMAX	>10 <sup>-7</sup>	5.E-5	Error criterion for checking consistency of redundant input data	R+4
RH (I) 5,8	(0,2)	1.E70	Fillet/round radius at end of Ith segment	R+4
XUSL(I) <sup>5</sup>	[0,∞)	1 . E70	)	R+4
ZUSL(I) <sup>5</sup>	(o, ∞)	1.E70	X and Z coordinates and slope (deg) of upper slicing plane. I≤10	R+4
TUSL(I) <sup>5</sup>	(-90°,+90°)	1.E70	<b>)</b>	R*4
XLSL(I) <sup>5</sup>	(0,0)	1.E70	)	R±4
ZLSL(I) <sup>5</sup>	[O, Ø)	1.E70	X and Z coordinates and slope (dcg) of lower slicing plane. I≤10	R+4
TLSL(I) <sup>5</sup>	(~90°,+90°)	1.E70	<b>\</b>	R+4
XSSL(I) <sup>5</sup>	(ه , ۵	1.E70	<b>)</b>	R+4
YSSL(I) <sup>5</sup>	(0,∞)	1.870	X and Y coordinates and slope (deg) of side slicing plane. I≤10	R+4
TSSL(I) <sup>5</sup>	(-90°,90°)	1 .E70	<b>)</b>	R+4

Namelist Variable	Value	Default Value	Meaning	TYPE <sup>7</sup>
RH∳US(I) <sup>5</sup>	(0,∞)	1.670	)Fillet/round radius on the upper, lower, and side slices,	R+4
rhøls(I) <sup>5</sup>	(0,∞)	1.E70	respectively. RHP—(I) fillets or rounds the beginning of	R+4
RH#SS(I) <sup>5</sup>	(0,∞)	1.E70	) segment I and the end of segment I-1, $2 < I < 10$ .	R+4
XSID(I) <sup>6</sup>	[0,2)		X and Z coordinates of side surface body line. Input only if body is asymmetrical and side surface	R+4
ZSID(I)	(-∞,∞)		body line is not in some horizontal plane as the geometric X axis. Defaults to X axis (see Fig. 8).	R+4
XMAPAX(I)	[0,∞)		X and Z coordinates of QUICK map axis (the axis relative to which the vehicle must be describable in	R+4
ZMAPAX(I)	(a,a_)		terms of a single-valued radius vector). Input only if body is asymmetrical and single-valued radius	R+4
			requirement cannot be met by default. Default to XSID, ZSID.	
TYPE	'SYMM'	'SYMM'	Vehicle is axially symmetric. Provide input for one body line.	R#4
	'ASYM'	·	Vehicle is asymmetric. Provide input for three body lines:	
			UCL, LCL, and SID (see Fig. 8). Input the needed combinations of CURVE, X, Z, TH. R, H, K, B.	
	,		DIR, and SLOPE in three successive &INPUT namelists. Provide slicing data, XSID, ZSID, XMAPAX,	
			ZMAPAX with any one of three if required. First of three namelists is upper centerline, second is	
			lower centerline, third is side.	-
IR <sup>10</sup>	[1,99]	5	Unit number for namelist input	I+4
IW <sup>10</sup>	[1,99]	6	Unit number for printed output	I+4
IF <sup>11</sup>	[1,99]	1	Unit number for output of QUICK input data	I+4
	≠IP		Unit IF will be rewound and printed on unit IW	
	-IP		Unit IF will not be rewound or printed. Punch QUICK input data	
IP	[1,99]	7	Unit number for card punch (need not be dummied if not used)	I+4
IPRINT	<b>≠</b> 0	o	Cards will be appended to QUICK input (unit IF) to exercise QUICK math model for checking	1+4
	<b>-</b> 0		No exercise cards generated	1

## Table 1. Concluded

Notes: 1'bbbb' - four blank characters. TH(I) and B(I) constitute segment equation for CURVE(I) -'LINE', i.e., Z(I) - X(I) \* TAN(TH(I)) + B(I). 3R(I). H(I). and K(I) constitute sogment equation for CURVE(I)-'\$GIV', i.e., (X(I)-H(I))\*\*2+(Z(I)-K(I))\*\*2 = R(I)\*\*2. <sup>4</sup>See Tables 3 and 4 for complete explanation. <sup>5</sup>Do not input these variables unless rounds, fillets, or slices are present.  $^{6}$ XSID(1) and ZSID(1) must both be input as zero. If input at all, at least two values of each must be input.  $^{7}$ Variable type: L = logical, I = integer, R = real. Number following \* is number of bytes. When defining a series of segments via X, Z, TH, R, H, K, and B, do not skip an I value when RH\*\*(I) is given. RH\*\*(I) will fillet/round the intersection of the Ith and I+1st segments. For example, to fillet/round two line segments, input X(1), Z(1), X(2), Z(2), and RH\*\*(1). Note further that each fillet/round reduces by one the total number of allowable segments: ten segments with two fillets will overrun QUICK's dimensions, whereas eight segments with two fillets will not. Though fillets/rounds are not treated as segments in the namelist input, they are treated as segments in the output after their intersections with adjacent segments have been computed and inserted into the XQ, ZQ, and TQ arrays. 9May be negative when TYPE -'ASYM' and if map axis is bent. 10 If the unit numbers are to be changed, they must also be changed in subroutine INIT because data are read from IR and written on IW before the namelist data are read as namelists. 11 If IF were set equal to IP(07) in the namelist, the input data would still be read from unit IR(05) copied to unit 01. Note: Character "#" means capital letter O "O" means numeric zero O [] means end points are included; () means end points are excluded.

Table 2. KWIKNOSE Input Options<sup>1</sup> for Three Main Body Lines (ZUCL, ZLCL, YSID)

CURVE(I)	X(I)	Z(I)	H(I)	K(I)	R(I)	SLOPE(I)	TH(I)	B(I)	DIR(I) <sup>3</sup>	May Follow Case 5, 9, 19?	ICASE(I) <sup>2</sup>	д. <sup>2</sup>
'\$GIV'	x	X	_	_	X	-		_	x	No	1	582
	x	x				x				Но	2 <sup>6</sup>	774
	x	<b> </b>	-		x	x			x	No	3 <sup>6</sup>	'834
	x	x			_		x	<del> </del> —		No	4	526
			x	x	x	_			x	Yes	5 <sup>4</sup>	624
	x	x			x		x		x	Yes	10 <sup>5</sup>	590
	x		x	х	x	_		_	x	Yes	11 <sup>5</sup>	626
į	_	x	x	х	I	_	_	_	x	Yes	12 <sup>5</sup>	628
			x	x	x	_	x	_	x .	Yes	13 <sup>5</sup>	632
	x	x	x	x	x		x			Yes	18 <sup>5,7</sup>	638
	_	-	_		x	x		. <del></del>	x	No	19 <sup>6</sup>	832
	x	x	<b>—</b> .	_	x	x		—	x	Yes	21 <sup>5,6</sup>	838
'LINE'	x	x	_	-		_	_			No	6	•
	x	_	-	_			x	_		No	7	10
		x		_			x			No	8	12
		_	_				x	x	x	Yes	9 <sup>4</sup>	136
	x	x	-				x			Yes	14 <sup>5</sup>	14
	x		:		_		x	x		Yes	15 <sup>5</sup>	138
		x	_		<b> </b>		٧.	x		Yes	16 <sup>5</sup>	140
	x	ж					x	¥	—	Yes	17 <sup>5,7</sup>	142
	x	x				x			x	No	206	262

L:	512 <sup>8</sup>	29	4	16	32	64	256	8	128

One of these options must be chosen for each segment. Insufficient or redundant data will cause execution to be halted.

<sup>&</sup>lt;sup>2</sup>ICASE(I) and L are <u>not</u> input variables but are used for program control. See subroutines CASE, SEGE, and INTE in Table 8.

<sup>&</sup>lt;sup>3</sup>DIR(I) is used only when multiple solutions or intersections can occur. When multiple roots are equal, DIR(I) is ignored. See Tables 3 and 4.

These options are not acceptable for last segment of vehicle.

<sup>4,5</sup> Use one of these options if previous segment was given via ICASE-5,9, or 19.

These options are not acceptable for first segment of vehicle.

<sup>&</sup>lt;sup>7</sup>These options contain redundant data, which if not self consistent to within ERRMAX, will cause execution to halted. <sup>8</sup>CURVE(I)=\*\*GIV\*\*, L=512; =\*\*LINE; L=0.

<sup>9</sup> If X(I) < 1.E70, L=2; IF X(I) ≥ 1.E70 (default), L=0. Same for remaining variables but with higher powers of 2.

AEDC-TR-77-89

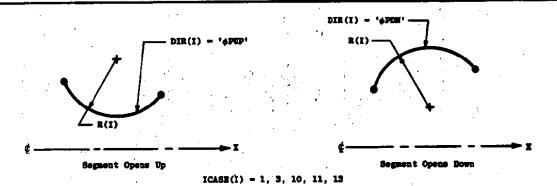
Table 3. Input Description for Variable DIR(I)<sup>1</sup>

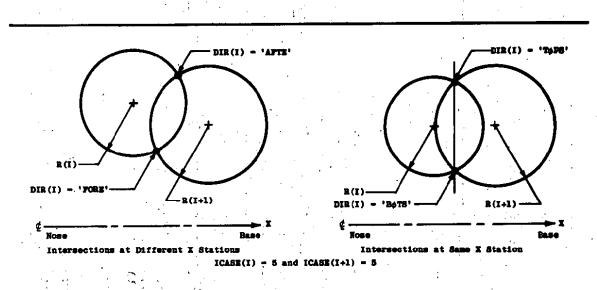
ICASE(I)	ICASE(I+1)	ICASE(I-1)	DIR(I)	Meaning
1, 3, 10, 11, 13			'oPUP' <sup>2</sup>	Ogive opens up
			' PDN ' 2	Ogive opens down
5	5		'PORE'	Use fore solution
			'AFTE'	Use aft solution
			'TOPS'	Use top solution   Use when X coordinate of fore
:			'BOTS'	Use bottom solution and aft solutions are equal
5	9	— <u>}</u>	'FORE'	Use fore solution
9	5	{	'AFTE'	Use aft solution
12			'FORE'	Use fore solution
·			'AFTE'	Use aft solution
20		5	'TOPS'	Use top solution
			'BOTS'	Use bottom solution
21		5	' <b>PUP</b> ' <sup>2</sup>	Ogive opens up
5	21	5	'APDN'2	Ogive opens down
21		9	. —-	DIR(I) not needed
. 19			'UPF¢'	Ogive opens up, use fore solution
			'UPAF'	Ogive opens up, use aft solution
			'UPT¢'	Ogive opens up, use top solution For DIR(I) notation, assume
			'UPB	Ogive opens up, use bottom solution ( 'UP'='\prup', 'DN'='\prup', 'DN'='
			'dnfo'	Ogive opens down, use fore solution ( 'Fo='FoRE', 'AF'='AFTE',
1			'DNAF'	Ogive opens down, use aft solution 'To'='ToPS', and 'Bo'='BoTS'
1			'DNT o'	Ogive opens down, use top solution and use Table 4.
			'DNB o'	Ogive opens down, use bottom solution

<sup>1</sup> For the ICASE(I) value, one of the indicated DIR(I) values must be chosen. See Table 4 for clarification.

 $<sup>^2</sup>$ Except for the noted values, DIR(I) refers to the aft intersection of the Ith segment.

Table 4. Nomenclature for Input Variable DIR(I)





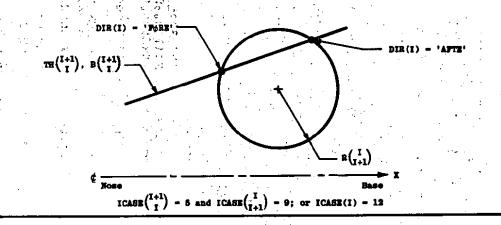


Table 5. Jobstream for Sample Case

```
/ PRIORITY
                                                          5
//VKF05333
                                                       JOB
                                                                                                      (XXX)
                  VRV00146+01+PROJ-NBR) + 100000 USER NAME++
//
                   MSGLEVEL= (2,0).
"
"
                   REGION=160K.
                  CLASS=X.
//
// TIME=(.15)
// EXEC FORTEPDS+PGMNO=VRV00146.
//GO.FT01F001 DD UNIT=WORK,DISP=(NEW,PASS,DELETE),DSN=&&WIKIN,
// DCB=(RECFM=F8+LRECL=80+BLKSIZE=800)+SPACE=(CYL+1)
//GO.FT05F001 DD *
HYPOTHETICAL ASYMMETRICALLY ABLATED. SLICED. BENT CONE
    LINPUT
    TYPE=!ASYM!.IPRINT=1,
    CURVE=2+'06IV',2+'LINE',DIR(2)='AFTE',
    H(1)=.5.K(1)=0.+R(1)=.5+TH(1)=45.+H(2)=-.64905+K(2)=1.14905+R(2)=1.125.
    X(3) = 4.92 \cdot Z(3) = 1.84 \cdot TH(3) = 10. \cdot RHO(2) = .5 \cdot X(4) = 9.22 \cdot Z(4) = 2.92218 \cdot TH(4) = 15. \cdot I
    LEND
    LINPUT
    CURVE=2+'OGIV',2+'LINE',DIR(2)='AFTE',
   H(1) = .375 + K(1) = 0.78(1) = .375 + TH(1) = 30.78886 + R(2) = 3.78886 + R(2) = 4.78886 
    X(3)=5.08242,Z(3)=1.88,TH(3)=10.,RHO(2)=.5,X(4)=9.22,Z(4)=2.24199,TH(4)=5..
   &END
    & INPUT
    CURVE=2+'OGIV',3+'LINE',DIR='OPDN','UPAF',
    H(1)=.4375.K(1)=0.,R(1)=.4375,TH(1)=36.325.SLOPE(2)=*T*,
   R(2) = 2.5625 + X(3) = 4.99909 + Z(3) = 1.8253 + RHO(2) = .5 + X(4) = 5.00333 + Z(4) = 1.8253 + RHO(2) = .5 + X(4) = 5.00333 + Z(4) = 1.8253 + RHO(2) = .5 + X(4) = 5.00333 + Z(4) = 1.8253 + RHO(2) = .5 + X(4) = 5.00333 + Z(4) = 1.8253 + RHO(2) = .5 + X(4) = 5.00333 + Z(4) = 1.8253 + RHO(2) = .5 + X(4) = 5.00333 + Z(4) = 1.8253 + RHO(2) = .5 + X(4) = 5.00333 + Z(4) = 1.8253 + RHO(2) = .5 + X(4) = 5.00333 + Z(4) = 1.8253 + RHO(2) = .5 + X(4) = 5.00333 + Z(4) = 1.8253 + RHO(2) = .5 + X(4) = 5.00333 + Z(4) = 1.8253 + RHO(2) = .5 + X(4) = 5.00333 + Z(4) = 1.8253 + RHO(2) = .5 + X(4) = 5.00333 + Z(4) = 1.8253 + RHO(2) = .5 + X(4) = 5.00333 + Z(4) = 1.8253 + RHO(2) = .5 + X(4) = 5.00333 + Z(4) = 1.8253 + RHO(2) = .5 + X(4) = .5
   X(5)=9.22.Z(5)=2.57165.TH(3)=10..
    XSID=0.,.85535,4.99909,5.00333,9.22,ZSID=0.,.02866,.02866.-.06865,.30026.
    XMAPAX=0..9.22,ZMAPAX=0..0..
   XUSL=6.86.9.65.ZUSL=3.11.1.83.
   XLSL=3.79.7.97.9.94.ZLSL=1.75.1.4..7.RHOLS(2) =4..
   XSSL=0..9.22.YSSL=2.3.2.3.
    &END
// EXEC FORTEPDS.PGMNO.VRV00099.COND=(8,LE)
//GO.FTOSFOO1 DD DSN=&&KWIKIN.DISP=(OLD.DELETE)
//GO.FT07F001 DD UNIT=3330.VOL=SER=666666.DSN=JCAG7115.
// DCB=(RECFN=U.BLKSIZE=80).DISP=(OLD.KEEP)
14
```

```
INPUT CARDS
INTPOTHETICAL ASYMMETRICALLY ABLATED. SLICED. BENT CONE
 LINPUT
 TYPE= *ASYM *. IPRINT=1.
 CURVE=2++0G1V+.2++LINE+.DIR(2)=+AFTE+.
 H(1)=,5,K(1)=0.,R(1)=,5,TH(1)=45.,H(2)=-,64905,K(2)=1.14905,R(2)=1.125,
 X(3)=4.92.Z(3)=1.84.TH(3)=10..RHO(2)=.5.X(4)=9.22.Z(4)=2.92218.TH(4)=15..
 LEND
 & INPUT
 CURVE=201061V1.201LINE1.DIR(2)=1AFTE1.
 H(1)=,375,K(1)=0.,R(1)=,375,TH(1)=30.,H(2)=-1.8125,K(2)=3,78886,R(2)=4.,
 X(3)=5.06242.Z(3)=1.68.TH(3)=10..RHO(2)=.5.X(4)=9.22.Z(4)=2.24199.TH(4)=5..
 LEND
 &INPUT
 CURVE=20.06IV.301LINE.DIR-OPDN..UPAF.
 H(1)=,4375.K(1)=0..R(1)=,4375,TH(1)=36,325.SLOPE(2)=+T+,
 R(2)=2.5625.X(3)=4.99909.Z(3)=1.8253.RHO(2)=.5.X(4)=5.00333.Z(4)=1.8253.
 x(5)=9.22.2(5)=2.57165..TH(3)=10..
 XSID=0.,.85535,4.99909,5.00333,9.22,ZSID=0.,.02866,,02866,-.06865,,30026,
 XMAPAX60..9.22.ZMAPAX=0..0..
 XUSL=6.86,9.65,ZUSL=3.11,1.83,
 XLSL=3.79.7.97.9.94.2LSL=1.75.1.4..7.RHOLS(2)=4..
 XSSL=0..9.22.YSSL=2.3.2.3.
I LEND
```

```
INPUT DATA! 1 ZUCL
                                                                                                             CURVE DIR SLOPE
                                                                5.00000E-01 9.9999E 69 9.99999E 69
                                                                                                              061V OPON
   9.99999E 69 9.99999E 69 4.50000E 01 5.00000E-01 0.0
                                                                                                              OGIV AFTE
               9.99998 69 9.99998 69 -6.49050E-01 1.14905E 00 1.12500E 00
2 9.9999E 69
                                                                                                              LINE OPDN
                                                                9.99999E 69
                           1.00000E 01 9.99999E 69
                                                    9.99999E 69
   4.92000E 00
               1-84000E 00
                                                                                                              LINE OPDN
                                                    9.99999E 69
                                                                9.99999E 69
                           1.50000E 01 9.99999E 69
   9.22000E 00
               2.92218E 00
                                                                                                                   OPON
                                                    9.9999E 69 9.9999E 69
                                                                            9.99998 69 9.99998 69
                            9.99999E 69
                                       9.99999E 69
   9.9999E 69
               9.99999E 69
                                                                                                                   OPDN
                                                                            9,9999€ 69 9,9999€ 69
                            9.9999E 69
                                       9,9999€ 69
                                                    9.99998 69 9.99998 69
   9.9999E 69
               9.99998 69
                                                                                                                   OPDN
                                                                9.999998 69 9.999998 69
                            9.99999E 69
                                       9.99999€ 69
                                                    9.99999E 69
               9.99999E 69
                                       9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69
                                                                                                                   OPON
                            9.9999E 69
   9.9999E 69
               9.99999E 69
                                                                                                                   OPDN
                           9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69
               9.99999E 69
                                       9.99998 69 9.99998 69 9.99998 69 9.99998 69 9.99998 69
                                                                                                                   OPDN
               9.99999E 69 9.99999E 69
10 9.9999E 69
LIMPUT1
                                                                        1.ERRMAX= 0.49999988E-04
                                                      7.1PRINT=
                          6. IF .
                                        1 . IP=
IRe
            1.14=
```

LEND

AEDC-TR-77-89

Table 6. Continued

```
NUMBER OF SEGMENTS NSEG . .
ICASE( 1) = 13
ICASE( 2) - 5
ICASE ( 3) = 14
ICASE ( 4) = 14
ICASE( 5) + 0
ICASE( 6) = 0
ICASE( 7) = 0
ICASE( 8) = 0
ICASE( 9) = 0
ICASE(10) = 0
SEGMENT END POINT COORDINATES AND SLOPES!
 1 0.0
                0.0
                             4.499998E 01 8.999899F 01
 2 1.464467E-01
                3.535535E-01
                            8.524072E 01
                                         4.500000E 01
   4.720713E-01
                1.055710E 00
                            9.99997E 00
                                         9.999997E 00
   4.919999E 00
                1.839999E 00
                            1.412625E 01
                                         1.412625E 01
   9.219999E 00
                2.922179E 00
                            9.9999948 69
   9.999994E 69
                9.999994E 69
                            9.999994E 69
                                         9.999994E 69
   9.999994E 69
                9.999994E 69
                            9.999994E 69
                                         9.999994E 69
   9.9999948 69
                9.999994E 69
                            9.999994E 69
                                         9.999994E 69
   9.999994E 69
                9.999994€ 69
                            9.9999946 69
                                        9.999994E 69
10 9.999994E 69
                9.9999948 69 9.9999948 69 9.9999948 69
11 9.99994E 69 9.999994E 69 9.999994E 69 9.999994E 69
SEGMENT EQUATIONS!
                                             R
   1.464467E-01 5.00000E-01 0.0
                                         5.000000F-01 9.999994E 69 4.500000E 01 9.999994E 69
   4.720713E-01 -6.490500E-01 1.149050E 00 1.125000E 00
                                                     9.99994E 69 9.999994E 69 9,999994E 69
   4.91999E 00 9.99994E 69
                            9.9999948 69 9.9999948 69
                                                     1.763268E-01 1.000000E 01 9.724715E-01
   9.219999E 00
                9.999994E 69
                            9.9999948 69 9.9999948 69
                                                     9.999994E 69
                                                                  1.500000E 01 4.516907E-01
   9.999994E 69 9.999994E 69
                            9.999994E 69
                                        9.999994E 69
                                                     9.999994E 69
                                                                  9.99994E 69 9.999994E 69
   9.999994E 69 9.999994E 69 9.999994E 69
                                                     9.999994E 69
                                                                  9-999994E 69
                                                                              9-99994E 69
   9.999994E 69 9.999994E 69 9.999994E 69
                                                     9.999994E 69
                                                                  9.999994E 69
   9.999994E 69 9.999994E 69 9.999994E 69 9.999994E 69
                                                     9.999994E 69
                                                                  9.999994E 69 9.999994E 69
   9.999994E 69 9.99994E 69 9.999904E 69 9.999994E 69 9.999994E 69 9.999994E 69
   9.999994E 69 9.999994E 69 9.999994E 69 9.999994E 69 9.999994E 69 9.999994E 69
FILLET/ROUND EQUATIONS AND INTERSECTIONS:
                  KSI
                             ETA
                                                     X1
                                                                 Z1
                                                                             71
                                                                                         x2
                                                                                                    Z2
                                                                                                                12
   1.46447E-01 9.99994E 69 9.9999E 69 9.99999E 69 9.99999E 69 9.99999E 69
                                                                         9.49999€ 69 9.99999€ 69
                                                                                                9.99999€ 69
                                                                                                            9.99999€ 69
               0-07869E-01 6.21313E-01
   4.72071E-01
                                      5.00000E-01 4.14969E-01 7.83694E-01
                                                                        7.10488E 01
                                                                                     0.01045E-01
                                                                                                1.11372E 00
                                                                                                            1.00000E 01
               9.99994 69 9.99996 69
   4-92000E 00
                                      9.999998 69
                                                 9.99999E 69
                                                             9.99999E 69
                                                                         9.99999E 69
                                                                                     9.99999€ 69
                                                                                                9.99999E 69
                                                                                                            9.99999€ 69
               9.9999E 69 9.9999E 69 9.9999E 69
   9.22000E 00
                                                 9.99999E 69
                                                             9.999998 69
                                                                         9.99999E 69
                                                                                     9.99999E 69
                                                                                                9.9999E 69
                                                                                                            9.9999E 69
   9.99999E 69
               9.99994E 69 9.99999E 69
                                      9.9999E 69
                                                 9-99999€ 69
                                                             9.99999E 69
                                                                         9.99999E 69
                                                                                     9.99999£ 69
                                                                                                9.99999E 69
                                                                                                            0.00000F 60
               9.99994E 69 9.99999E 69 9.99999E 69
                                                 9.99999E 69 9.99999E 69
                                                                         9.99999E 69
                                                                                     9.9999E 69
                                                                                                9.9999E 69
                                                                                                            9.9999E 69
   9.99999E 69
               9.99999E 69
                          9.9999E 69
                                     9.9999E 69
                                                 9.99999E 69 9.99999E 69
                                                                         9.999998 69
                                                                                     9,9999€ 69
                                                                                                9.99999 69
                                                                                                            9.99999€ 69
   9.9999E 69
               9.99994E 69
                                      9.99998 69 9.99998 69 9.99998 69
                          9.9999E 69
                                                                         9.9999E 69
                                                                                     9.9999E 69
                                                                                                9,99999€ 69
                                                                                                            9.99999E 69
   9.99999£ 69
               9.99999E 69
                          9.99999E 69
                                                 9.999998 69 9.999998 69 9.999998 69 9.999998 69
                                      9.99999E 69
                                                                                                            9.99999E 69
10 9.99996 69 9.99996 69 9.99996 69 9.99996 69 9.99996 69 9.99996 69 9.99996 60 9.99996 60 9.99996 60 9.99996 69
                                                                                                            9.99999€ 69
```

Table 6. Continued

```
FINAL QUICK ARRAYS
                              4.499998E 01 8.999899E 01
 2 1.464467E-01 3.535535E-01 7.104884E 01 4.500000E 01
 3 4.149692E-01 7.836936E-01 1.000000E 01 7.104884E 01
 4 8.010449E-01 1.113717E 00 9.999997E 00 1.000000E 01
 5 4.919999E 00 1.839999E 00 1.412625E 01 1.412625E 01
 6 9.219990E 00 2.922179E 00 9.999994E 69 9.999994E 69
 7 9.99994E 69 9.99994E 69 9.99994E 69 9.99994E 69
 8 9,99994E 69 9,99994E 69 9,999994E 69 9,999994E 69
 9 9.99994E 69 9.999994E 69 9.999994E 69 9.999994E 69
10 9.909994E 69 9.909994E 69 9.999994E 69 9.999994E 69
IMPUT DATA: 2 ZLCL
                               TH "
                                                                                                             CURVE DIR SLOPE
 1 9.9999E 69 9.9999E 69 3.0000E 01 3.75000E-01 0.0
                                                                3.75000r-01 9.99999E 69 9.99999E 69
                                                                                                              OGIV OPON
 2 9.99999E 69 9.99999E 69 9.99999E 69 -1.81250E 00 3.78886E 00 4.00000E 00 9.99999E 69 5.00000E-01
                                                                                                              OGIV AFTE
3 5.0824ZE 00 1.88000E 00 1.00000E 01 9.99999E 69 9.99999E 69 9.99999E 69
                                                                            9.99999E 69
                                                                                        9.99999€ 69
                                                                                                             LINE OPON
 4 9.22000E 00
               2.24109E 00 5.00000E 00 9.99909E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69
                                                                            9.99999E 69
                                                                                        9,9999€ 69
                                                                                                             LINE OPON
 5 9.99999E 69
                                                                            9,9999E 69
                                                                                        9.99999£ 69
 6 9.99998 69 9.99998 69 9.999998 69 9.999998 69 9.999998 69 9.999998 69
                                                                                        9.9999E 69
7 9.99998 69 9.99998 69 9.99998 69 9.99998 69 9.99998 69 9.99998 69 9.99998 69
                                                                                        9.99999E 69
 8 9.99999E 69 9.99999E 69
9 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.9999E 69 9.99999E 69 9.99999E 69
10 9,99995 69 9,99996 69 9,99995 69 9,99995 69 9,99995 69 9,99995 69 9,99995 69 9,99995 69
&INPUT1
                                       1,10=
IA-
            1.Ive
                                                      7.1PRINT=
                                                                       1.ERRMAX= 0.4999988E-04
LEND
NUMBER OF SEGMENTS NSEG # 4
ICASE( 1) = 13
ICASE( 2) = 5
ICASE( 3) = 14
ICASEL 41 = 14
ICASE( 5) = 0
ICASE( 6) = 0
ICASE( 7) =
ICASE( 6) =
ICASE( 9) =
ICASE(10) =. 0
SEGMENT END POINT COORDINATES AND SLOPES!
                             2.99998E 01 8.999899F 01
                0.0
 2 1.875001E-01 3.247595E-01 4.971005E 01 3.000002E 01
3 1.238627E 00 1.202235E 00
                             9.99997E 00 9.99997E 00
                1.679999E 00
                             4.999980E 00 4.999980E 00
 4 5.082419E 00
   9.219999E 00 2.241989E 00
                             9.99994E 69 9.99994E 69
```

9,99994E 69 9,999994E 69 9,999994E 69 9,999994E 69 9,999994E 69

AEDC-TR-77-89

Table 6. Continued 8 9.99994E 69 9.999994E 69 9.99994E 69 9.999994E 69 9.999994E 69 9.99994E 69 9.999994E 69 9.999994E 69 9.99994E 69 9.99994E 69 9.999994E 69 9.999994E 69 11 9.999994E 69 9.9999948 69 9.9999948 69 9.9999948 69 SEGNENT EQUATIONS: ĸ 3.750000f-01 9.999994E 69 3.000000E 01 9.999994E 69 1.875001E-01 3.750000E-01 0.0 1.238627E 00 -1.812500E 00 3.788659E 00 4.000000€ 00 9.99994E 69 9.999994E 69 9.999994E 69 9.99994E 69 9.999994E 69 5.082419E 00 9.999994E 69 1.763268E-01 1.000000E 01 9.838324E-01 9.999994E 69 9.999994E 69 9.9999948 69 9.99994E 69 5.000000E 00 1.435345E 00 9,999994E 69 5 9.999994E 69 9.999994E 69 9.999994E 69 9.999994E 69 9.999994E 69 9.999994E 69 9,999994E 69 9.9999948 69 9.9999948 69 9.999994E 69 9.99994E 69 9.99994E 69 9.999994E 69 9.999994E 69 9.999994E 69 9.909994E 69 9.999994E 69 9.99994E 69 9.999994E 69 9.999994E 69 10 9.999994E 69 FILLET/ROUND EQUATIONS AND INTERSECTIONS: KSI ETA RHO X1 21 Ti XZ. ZZ TZ 1 1-87500E-01 9.4<del>99</del>99E 69 9.99999£ 69 9.99999£ 69 9.99999£ 69 9.99999£ 69 9.99999E 69 9.99999E 69 1.23863E 00 1.49739E 00 7.40149E-01 1.07890E 00 4.73521E 01 5.00000E-01 1.12962E 00 1.41057E 00 1.232558 00 5.08242E 00 9.99999E 69 9.9999E 69 9.9999E 69 9.99999E 69 9.99999E 69 9.9999FE 69 9.99999E 69 9.99999E 69 9.99999E 69 9.22000E 00 9.99994E 69 9.99999E 69 9.99999E 69 9.999998 69 9.99999E 69 9.99999£ 69 9.9999E 69 9.9999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.999998 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9,9999E 69 9.99999E 69 9.99999E 69 9.9999E 69 9.999998 69 9.9999E 69 9.99999E 69 9.9999£ 69 9.99999E 69 9.9999E 69 9.99999E 69 9.99999E 69 9.99999£ 69 9.9999E 69 9-9999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.9999E 69 9.999998 69 9.99999E 69 9.99999E 69 9.99999€ 69 8 9.9999E 69 9.99998 69 9.99998 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.999998 69 9.999996 69 9.999996 69 9.999996 69 9.999996 69 9.999996 69 9.99999E 69 9.99999E 49 9.99999E 69 9.99999E 69 10 9.9999E 69 9.99999£ 69 9.99999£ 69 9.99999£ 69 9.99999£ 69 9.99999£ 69 9.9999E 69 9.9999E 69 FINAL QUICK ARRAY: · TA TF 1 0-0 0.0 2.999998E 01 8.999899E 01 1.875001E-01 3.247595E-01 4.735210E 01 3.000002E 01 3 1.129624E 00 1.078896E 00 1.000000E 01 4.735210E 01 1.410566E 00 1.232553E 00 9.99997E 00 1.00000E 01 5.082419E 00 1.879999E 00 4.999980E 00 4.999980E 00 9.219999E 00 2.241989E 00 9.9999948 69 9.9999948 69 9.999994E 69 7.99994E 69 9.99994E 69 9.99994E 69 9.99994E 69 10 9.999994E 69 9.999994E 69 9.999994E 69 9.999994E 69 INPUT DATA: 3 YSID 1 Z CURVE DIR SLOPE ----------1 9.99999E 69 9.99999E 69 3.63250E 01 4.37500E-01 0.0 4.375004-01 9.499998 69 9.99999E 69 OGIV OPDN 2 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 2.56250E 00 9.99999E 69 5.00000E+01 OGIV UPAF 3 4.9990F 00 1.82530E 00 1.00000E 01 9.9999E 69 9.9999E 69 9.9999E 69 9.9999E 69 9.99999E 69 LINE OPDN F

9.99999E 69

LINE OPDN

LINE OPDN

4 5.00333E 00 1.82530E 00 9.99999E 69 9.99999E 69 9.99999E 69 9.9999E 69 9.99999E

5 9.22000E 00 2.57165E 00 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69

ZUS	L X-SL		Z-SL	_			RHO-S
1	6.86000E		3.11000E		9,99999	69	9,99999E 69
2	9.65000E		1-83000E		9.99999E		9,99999E 69
3	9.99999E		9.99999E		9.99999E		9.999998 69
3	9.99999E		9.99999E		9.99999E		9.99999E 69
6	9.9999E		9.99994E		9.99999E		9.99999E 69
7	9.9999E		9.9999E		9.9999E		9,99999E 69
8	9.9999E		9-9999E		9.9999E		9.99999 <u>E</u> 69
	9.99998		9-99999		9.9999E		9.99999£ 69
10	9.9999E	69	9•9999E	69	9.99999E	69	9.99999E 69

ZLSI	ZLSL X-SL		Z-SL	T-SL		RHO-S
				******	<b></b>	
1	3.79000E	00	1.75000E 00	9.9999E	69	9.99999E 69
2	7.97000E	00	1.40000E 0	9.9999E	69	4.00000E 00
3	9.94000E	00	7.00000E-0	9.99999E	69	9.99999E 69
4	9,9999E	69	9.9999E 69			9.99999E 69
5	9.9999E	69	9.99999E 69	9.9999E	69	9.99999E 69
6	9.9999E	69	9.99999E 69			9.99999E 69
7	9.99999E	69	9.99999E 69			9.99999E 69
à	9.99999£	69	9.99999E 69			9.99999E 69
9	9.9999E	69	9.99999E 69			9.99999E 69
10	9.9999E	69	9-9999E 69			9.99999E 69

YSS			2-SL		T-SL		RHQ-S
				***			
1	0.0		2-30000E	00	9.99999E	69	9.99999E 69
2	9.22000E	00	2.30000E	00	9.99999E	69	9.99999E 69
3	9.99999E	69	9.999998	69	9.9999E	69	9.99999E 69
4	9.99999E	69	9.99999€	69	9.9999E	69	9.99999E 69
5	9.99999	69	9.99999E	69	9.9999E	69	9.99999E 69
6	9.9999E	69	9.99999E	69	9.99999E	69	9.99999E 69
7	9.9999E	69	9.9999£	69	9.99999E		9.9999E 69
8	9.9999E	69	9-9999E	69	9.99999E	69	9.99999E 69
9	9.9999E	69	9.99999E		9.9999E		9.9999E 69
10	9.9999E		9.9999E		9.9999E		9.9999E 69

	XMAPAX		ZHAPAX	XSID		ZS10	
1	0.0		0-0		0.0		0.0
2	9.22000E	00	0.0		8.55350E-	-01	2.86600E-02
3	9.9999E	69	9.99999E	69	4.99909E		2.86600E-02
4	9.9999E	69	9.99999E	69	5.00333E		-6.86499E-02
5	9.9999E	69	9.99999E	69	9.22000E		3.00260E=01
6	9.9999E	69	9.9999E	69	9.99999E		9.9999€ 69
7	9.9999E	69	9.99999E	69	9.9999E	69	9.99999E 69
8	9.99999E	69	9-99999E	69	9.9999E		9.9999E 69
9	9.9999E	69	9-99999E	69	9.99999E	69	9.99999E 69
10	9.9999E	69	9.9999E	69	0.0000F	40	9.99999F 49

Table 6. Continued

Lini ir= Leni	•	1.1	/= 	6	•IF=	1,IP=	7	.IPRINT=		1.ERR	4AX=	0.499999	88E-04		
			's NSEG				••••			•••••			••••••	•••••	•••••
-			12 M2En												
	iE ( 1) iE ( 2)														
ICAS	E ( 3)	= 14													
	E ( 4)														
	iE ( 5) iE ( 6)														
ICAS	E( 7)	- 0													
	iE ( 8) iE ( 9)														
	E (10)														
ė F Či	ENT EL			<b></b>	AND SLOPES:										
3E OF															
	×		Z		TA	TF 									
1	0.0		0.0		3.632498E	01 8.999899:									
		75E-01			1 5.893269E 0 9.999994E										
		89E 00	1.8252			00 9.999994E									
		29E 00	1.8252	99E 0	0 1.003739E	01 1.003739E									
		99E 00	2.5716 9.9999												
		94E 69	9.9999												
		94E 69			9 9.999994E										
		94E 69			9 9.999994E	69 9.999994E									
	-					0, ,,,,,,,,	•								
SEGF	IEMI EG	UATION!	; t												
	×		H		K	R 		M		T		8			
1	1.7834	06E-01	4.3750	00E-0	1 0.0	4.375000=		9.99994E		J.632500E					
					0 - 2.417009E			9.999994E		9.999994E		9.99999	4E 69		
		89E 00	9.9999		9 9,999994E 9 9,999994E			1.763268E	-01	1.0000008	01				
		99E 00	9.9999					1.769999E	-01	1.003739E	. 01	1.82529			
		94E 69			9 9.999994E			9,99994E	69	9.999994E	69	9.9999	4E 69		
•		94E 69	9,9999					9.999994E		9,999994E		9,99999			
9	9.9999	94E 69	9.9999	94E 6	9 9.999994E	69 9.999994E	69	9.999994E	69	9.999994E		9,99999			
10	9.9999	94E 69	9.9999	94E 6	9 9. <b>9</b> 99994E	69 9.999994E	69	9.99994E	69	9.999994	69	9.99999	4E 69		
FILL	ET/ROU	ND EQU	TIONS A	ND IN	TERSECTIONS:										
*	X		KSI		ETA	RH0		x1		<b>?</b> 1	1	1	X2	zz	ST
1	1.7834	1E-01	9-99999	E 69	9.99999E 69	9.99999E 69	9.9						9.99999E 69		9.99999E 69
	8.5534 4.9990		1-15439		6.39662E-01								1.06756E 00		1.00000E 01
	5.0033		9.99994		9.99999£ 69								9.99999E 69 9.99999E 69		9.99999E 69
5	9.2200	0E 00	9-99994	E 69	9.9999E 69	9.9999E 69							9.99999E 69		9.99999E 69
	9.9999		9.999991		9.99999E 69			9999E 69	9.999	99E 69 9	.999	99E 69	9.999 <del>9</del> E 69	9.9999E 69	9.99999E 69
7	9.9999	TE OY	9.99994	. 69	9.99999E 69	9.9999E 69	9.9	9999E 69	9.999	999E 69 9	.999	99E 69	9 <b>.99999£</b> 69	9.99999E 69	9.99999E 69

Table 6. Continued

8 9.99998 69 9.9998 69 9.99998 69 9.99998 69 9.99998 69 9.99998 69 9.99998 69 9.99998 69 9.99998 69 9.99998 69 9.99998 69 9.99998 69 9.99998 69 9.99998 69 9.9998 69 9.99998 69 9.99998 69 9.99998 69 9.99998 69 9.99998 69 9.99998 69 9.99998 69 9.99998 69 9.99998 69 9.99998 69

FINAL QUICK ARRAY:

Z X 3.632498E 01 8.999899E 01 2 1.783406E-01 3.524806E-01 5.452428E 01 3.632498E 01 3.7472076E-01 9.298420E-01 1.000000E 01 5.452428E 01 1.132066E 00 9.999994E 00 1.000000E 01 4 1.067565E 00 5 4.999089E 00 1.825299E 00 0.0 0.0 1.003739€ 01 1.825299E 00 1.003739E 01 6 5.003329E 00 7 9.219999E 00 2.571650E 00 9.999994E 69 9.999994E 69 9,99994E 69 9,999994E 69 9,999994E 69 8 9.999994E 69 9 9.999994E 69 9.999994E 69 9.999994E 69 9.999994E 69 10 9,999994E 69 9.999994E 69 9.999994E 69 9.999994E 69

SLICE GEOMETRY!

SIDE LOWER UPPER Ŧ Ż 2.067344E 00 -4.786331E 00 2.299999E 00 0.0 0.0 6.257238E 00 -2.464478E 01 2.299999E 00 7,96999E 00 1,40000E 00 -4.786331E 00 9.219999€ 00 0.0 1.830000E 00 -2.A64478E 01 2 9.650000E 00 9.999994E 69 9.999994E 69 9,999994E 69 9.940000E 00 7.000000E-01 -1.956165E 01 9.99994E 69 9.999994E 69 3 9.999994E 69 9,999994E 69 9,999994E 69 9.99994E 69 9.99994E 69 9.999994E 69 9.999994E 69 4 9,999994E 69 9.99994E 69 9.999994E 69 9.999994E 69 9.99994E 69 9,999948 69 9,9999948 69 9.999994E 69 9.99994E 69 9.99994E 69 5 9.999994E 69 9.999994E 69 9,999994E 69 9,999994E 69 9.999994E 69 9.999994E 69 9,999994E 69 6 9.999994E 69 9,999948 69 9,9999948 69 9.999994E 69 9.999994E 69 9.999994E 69 9.999994E 69 9.999994E 69 9.999994E 69 7 9.999994E 69 9.99994E 69 9.99994E 69 9.99994E 69 9.999994E 69 9.999994E 69 9,99994E 69 9,99994E 69 9,999994E 69 9.99994E 69 9.999994E 69 9.999994E 69 9.999994E 69 9.999994E 69 9.99994E 69 9.999994E 69 9.999994E 69 9.999994E 69 9.99994E 69 9.999994E 69 9.999994E 69 9,99994E 69 9,99994E 69 9.999994E 69 9,99994E 69 9,99994E 69 9.99994E 69 10 9.99994E 69 9.999994E 69 9.999994E 69 SLICE ROUND/FILLET CENTERS AND INTERSECTIONS: 12 Tì X2 X1 21 81 ĐJ

22 -1.94665E 00 -1.30367E-02 -2.54278E 00 7.11942E 00 7.45318E 00 1.44327E 00 -4.78633E 00 8.45870E 00 1.22635E 00 -1.95616E 01

FINAL QUICK ARRAYS FOR ROUNDED/FILLETTED SLICES:

x 2		
1 0.0	94E 69 9,999994E 6 94E 69 9,999994E 6	0.0 9.999994E 69 9.999994E 69 9.999994E 69 9.999994E 69 9.999994E 69 9.999994E 69

LISTING OF QUICK IMPUT DECK ON UNIT 11

AEDC-TR-77-89

Table 6. Continued

						+		
INYPOTHET	ICAL AS	YMMETR	ICALLY AB	LATED. SLI	CED. BENT	CONE		
1 1								į
1 6  LELL	1ELLI	PIEC	CROSS SE	CTION SIO	LSCP			
IUELL	SELLI	PIEC	SID	TUCL	USCP			
ILBSL	3L INE	PIEC	BLSL	LSLCP			LELL	
ILSSL	<b>4LINE</b>	PIEC	LSLCP	SSL		LELL		i
IUSSL	SLINE		SSL	USLCP			UELL	į
JUTSL I 1	6L INE MAP	PIEC	USLCP	TUSL		UELL		
iiı	0,	. 0	9.22000		•			
FYUCL								ï
I I LINE	PIEC							i
0.0  -1	0.	Q	9.22000	. 0.0 A	0.0 A	0.0		į
IZUCL								!
I 1 ELLX	PIEC	KVO						:
1 0.0	0.		0.14645	0.35355A	89.99899A	44.99998		i
1 2 ELLX	PIEC	KVO						ı
1 0.146 1 3 ELLX	PIEC.	35355 KV0	0-41497	0./5369A	45.00000A	71.04864		. !
0.414		78369	0.80104	1.11372A	71.048844	10.00000		:
I 4 LINE	PIEC					*		i
0.801		11372	4.92000	1.84000A	10.00000A	10.00000		į
1 5 LINE 1 4.920	PIEC	84000	0.22000	2 622184	14 124254	14 12425		. !
i-1	**	04000	*155000	2.92218A	144150524	14.15052		
ZLCL								i
1 ELLX	PIEC							i
I 2 ELLX	PIEC		0.18750	-0.32476A	-89 <b>.9</b> 98 <b>99</b> A	-29.99998		ļ
	50 -0.		1.12962	-1.07890A	30.000024	-47-35210		
I 3 ELLX	PIEC		******	-20010702	-304404EM	-4.433210		- :
1.129		07890	1.41057	-1.23255A-	47.35210A	-10.00000		i
4 LINE   1.410	P1EC 57 -1.		E 00343					
I S LINE	PIEC		3+48545	-1.88000A-	-10-00000A	-10-00000		
	42 -1.		9.22000	-2.24199A	-4.99998A	-4.99998		:
1-1								i
IYSID   1 ELLX	PIEC	wwa						
0.0	PIEC O.	K¥Q	0.17834	0.352ARA	89.99899A	36.32498		!
I Z ELLX	PIEC		*******	*******	07,770772	30.32470		- :
1 0.178		35248	0.74721	0,92984A	36.32498A	54.52428		i
1 3 ELLX	PIEC							i
† 0.747;   4 LINE	PIEC.	92984	1.06756	1.13207A	54.52428A	10.00000		. !
1.067		13207	4.99909	1.825304	10.00000A	9.99999		!
I 5 LINE	PIEC					*******		- 1
1 4.999		82530	5.00333	1.82530A	0.0 A	0.0		i
6 LINE   5.003	PIEC		0 33000	2 621464				
-1	<b>33 10</b>	82530	*******	2,57165A	10.03/39A	10.03739		Į.
İZÜSL								!
1 LINE	PIEC							i
1 0.0	0.	25724	9.65000	1.03000A-	Z4.6447BA	-24.64478		i
I-1 IZLSL								į
I LINE	PIEC	KV5						!
1 0.0		06734	7.45318	-1.44327A	4.78633A	4.78633		į

Table 6. Concluded

1 2 ELLX						
1 7.453	18 -1.44327	. 8.45870	+1.22635A	4.78633A	19.56165	
1 3 LINE	PIEC KV5					
8.4581	0 -1.22635	9.94000	-0.70000A	19.56165A	19.56165	•
1-1					• .	
I YSSL	10					
	PIEC KVS					
0.0	, 2.30000	9.55000	2.30000A	0.0 A	0.0	
I <b>-</b> 1						
ZSID						
	PIEC KVS					
0.0	0.0	0.65535	0.02866A	1.91908A	1.91908	
	PIEC KV5					
0.0553		4.99909	0.02866A	0.0 A	0.0	
3 LINE	PIEC KVS					•
4.9990		5.00333	-0.06865A	-87.50502A	-87.50502	
I 4 LINE	PIEC KVS	•	•			
	13 -0.06865	9.22000	0.30026A	4,999994	4.99999	
t-1 ·						
ZMAP	b			•	-	
1:1 LINE :	PIEC KVS		•			
0.0	0.0	9.22000	, 0.0 A	0.0 A	0.0	
<b>-1</b> .						
I YUSL	YUCL		. / ·			
YLSL	YUCL			•		
YLCL	· YUCL	1 .		• 1	• .	
YHAP	YUCL		* .		** ** ** ***	
YUSÇP	YSID			•		
YLSCP	A210					
YUSLCP	YSSL					
YLSLCP	YSSL					
ZUSCP	ZUCL				•	
ZLSCP	ZLCL					
ZUSLCP	ZUSL					
ZLSLCP	ZLSL					
IZSSL	2510	٠,		5, 3, 2		
, ,			4	•		
	4 44144	0 22100				

# Table 7. Definitions of Logical Names for QUICK Cross-Section and Body Line Models

### Cross-Section Models

Logical Name	Definition
`	
LELL	Lower ellipse
UELL	Upper ellipse
UTSL	Upper top slicing plane
USSL	Upper side slicing plane
LSSL	Lower side slicing plane
LBSL	Lower bottom slicing plane
	Body Line Models
LCL	Lower centerline
LSCP	Lower slope control point
MAPAX	Map axis
SID	Side
UCL	Upper centerline
USCP -	Upper slope control point
LSLCP	Control point for lower horizontal and vertical
	slicing planes
LSL	Centerline on lower slice
SSL	Side on side slice
USL	Centerline on upper slice
USLCP	Control point for upper horizontal and vertical

Note: See Fig. 8 also.

slicing planes

	.35	•	
	ietala • mari		
		PUT FOR AN AXISHMETRIC STACK OF	
	S. NOT NECESSARILY TAN	GENT	
ILLINAL PROGRA	M VERSION 2/18/77		_
PUT INSTRUCTI	ONS (SEE AEDC-TR-77-89 FOR		-
	D IS ID DATA (BOAL-QUI		
	IS VIA NAMELIST /INPU		
	TO MAKE QUICK EXERCISE		
		= 1) WITH DCB=(RECFM=FB+	
	L LONG SEQUENTIAL FILE		
	INPUT DECK WILL BE WRI	TTEN ON IP WITHOUT REWINDING. US	šΕ
PUNCH.			
		0}•K(10)•R(10)•B(10)•M(10)•	
		Z1(10).Z2(10).T1(10).T2(10).	
		Q(11)+TQ(11+2)+KRV(3)+CL(2)+	
	.3.3) .SL (3) .RHOSL (10.3		
		10) +ZMAPAX(10) +XSID(10) +ZSID(10)	
5 BLB(8)			
		ASE (7) +NP (3) +NS (3) +NB (8)	
		LNK+RINF+ERRHAX+OPDN+ELLX+	
		UP-OGIV-PIO180-XJ-ZJ-XI-ZI-HI-	
	UPF0+UPAF+UPT0+UPB0+DNI	THIJ+SIJ+AI+HJ+RJ+A++DX	
COMMON ATEA	10.1w.1f.10.1P&INT.NCF	G.NSEGP1.ISTOP.JSTOP.KSTOP.	
	LM1+I1+NPTS+IC+ISL+LB+		
1 1000661616			
		D(80)	
COMMON /LA/	SLOPE(10), IDEN(80).CAR		٠
COMMON /LA/	SLOPE (10) , IDEN (80) . CAR		٠
COMMON /LA/ COMMON /LS/ REAL K+M+KS LOGICAL+1 S	SLOPE(10), IDEN(80).CAR	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K.M.KS LOGICAL*1 SI LB=1	SLOPĒ(10),IDEN(80).CARI TI.T I.KI.MI.KJ.MJ.MIJ.LINE	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K+M+KS LOGICAL+1 SI LB=1 CALL INIT	SLOPĒ(10),IDEN(80).CARI TI.T I.KI.MI.KJ.MJ.MIJ.LINE	•KV•KRV	•
COMMON /LA/ COMMON /LS/ REAL K.M.KS LOGICAL*1 S LB=1 CALL INIT CALL INPT	SLOPE(10),IDEN(80).CARI TI-T I-KI-MI-KJ-MJ-MIJ-LINE LOPE,IDEN,CAHO-TI-T	•KV•KRV	
COMNON /LA/ COMHON /LS/ REAL K+M+S/ LOGICAL+1 S LB=1 CALL INIT CALL INPT IF (TYPE+EQ+	SLOPĒ(10),IDEN(80).CARI TI.T I.KI.MI.KJ.MJ.MIJ.LINE	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K+M+KS LOGICAL+1 S LB=1 CALL INIT CALL INPT IF(TYPE-EQ- CALL CASE	SLOPE(10),IDEN(80).CARI TI-T I-KI-MI-KJ-MJ-MIJ-LINE LOPE,IDEN,CAHO-TI-T	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K+M+KS LOGICAL+1 SI LB=1 CALL INIT CALL INPT IF (TYPE-EQ- CALL CASE CALL SEGE	SLOPE(10),IDEN(80).CARI TI-T I-KI-MI-KJ-MJ-MIJ-LINE LOPE,IDEN,CAHO-TI-T	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K+M+KS LOGICAL+1 SI LB=1 CALL INIT CALL INPT IF(TYPE-EQ- CALL CASE CALL SEGE CALL INTR	SLOPE(10), IDEN(80), CARI TIST ISKISMISKJSMJSMIJSLINE LOPE, IDEN, CARDSTIST ASYM) GO TO 100	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K+M+KS LOGICAL*1 SI LB*1 CALL INPT IF(TYPE.EQ. CALL CASE CALL SEGE CALL INTR CALL INTR	SLOPE(10), IDEN(80), CARI TIST ISKISMISKJSMJSMIJSLINE LOPE, IDEN, CARDSTIST ASYM) GO TO 100	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K+M+KS LOGICAL+1 SI LB=1 CALL INIT CALL INPT IF(TYPE-EQ- CALL CASE CALL SEGE CALL INTR	SLOPE(10), IDEN(80), CARI TIST ISKISMISKJSMJSMIJSLINE LOPE, IDEN, CARDSTIST ASYM) GO TO 100	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K.M.KS LOGICAL*1 S LB=1 CALL INPT IF(TYPE.EQ. CALL CASE CALL CASE CALL SINTR CALL SINTR CALL GARY CALL PRNT CALL PRNT CALL SLCE	SLOPE(10), IDEN(80), CARI TIST ISKISMISKJSMJSMIJSLINE LOPE, IDEN, CAROSTIST ASYM) GO TO 100	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K+M+KS LOGICAL+1 SI LB=1 CALL INIT CALL INPT IF (TYPE.EQ. CALL CASE CALL SEGE CALL INTR CALL GARY CALL PROT CALL PROT	SLOPE(10), IDEN(80), CARI TIST ISKISMISKJSMJSMIJSLINE LOPE, IDEN, CAROSTIST ASYM) GO TO 100	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K+M+KS LOGICAL+1 SI LB=1 CALL INIT CALL INPT IF (TYPE-EQ- CALL SEGE CALL SEGE CALL INTR CALL GARY CALL PRNT CALL ROFL CALL SCCE CALL ROFL CALL SCCE CALL SLCE CALL SCCE CALL GALY	SLOPE(10), IDEN(80), CARI TIST ISKISMISKJSMJSMIJSLINE LOPE, IDEN, CAROSTIST ASYM) GO TO 100	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K+M+KS LOGICAL*1 SI LB*1 CALL INPT IF (TYPE.EQ. CALL CASE CALL SEGE CALL SEGE CALL SEGE CALL PARY CALL PARY CALL POFL CALL SLCE CALL SLCE CALL SLCE CALL SLCE CALL SLCE CALL SLCE CALL SLCE CALL SLCE	SLOPE(10), IDEN(80), CARI TIST ISKISMISKJSMJSMIJSLINE LOPE, IDEN, CAROSTIST ASYM) GO TO 100	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K+M+KS LOGICAL*I S LB*I CALL INPT IF ITYPE.EQ. CALL CASE CALL SITE CALL SITE CALL SITE CALL PRNT CALL PRNT CALL SLCE CALL SLCE CALL SLCE CALL LIST CALL SLCE CALL LIST CALL SLCE CALL LIST CALL LIST STOP	SLOPE(10), IDEN(80), CARI TIST ISKISMISKJSMJSMIJSLINE LOPE, IDEN, CARDSTIST ASYM) GO TO 100	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K+M+KS LOGICAL+1 SI LB=1 CALL INIT CALL INPT IF (TYPE-EQ- CALL SEGE CALL SEGE CALL INTR CALL INTR CALL INTR CALL PRNT CALL ROFL CALL SCE CALL SLCE CALL FSL CALL LIST STOP DO 00 120 LB=1	SLOPE(10), IDEN(80), CARI TI-T TI-T I-KI-MI-KJ-MJ-MIJ-LINE LOPE, IDEN, CAHO-TI-T ASYM) GO TO 100	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K.M.KS LOGICAL*1 SI LB*1 CALL INIT CALL INPT IF (TYPE.EQ. CALL SEGE CALL SI CALL SUCE CALL SUCE CALL SI CAL	SLOPE(10), IDEN(80), CARI TI-T TI-T I-KI-MI-KJ-MJ-MIJ-LINE LOPE, IDEN, CAHO-TI-T ASYM) GO TO 100	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K.M.KS LOGICAL*I SI LB*I CALL INIT IF(TYPE.EQ. CALL SEGE CALL INTR CALL INTR CALL GARY CALL PRNT CALL ROFL CALL SLCE CALL RFSL CALL GUCK CALL LIST STOP DO 0120 LB*I IF(L8.EQ.1) CALL INIT	SLOPE(10), IDEN(80), CARI TI-T TI-T I-KI-MI-KJ-MJ-MIJ-LINE LOPE, IDEN, CAHO-TI-T ASYM) GO TO 100	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K+M-KS LOGICAL+1 SI LB=1 CALL INIT CALL INPT IF(TYPE-EQ- CALL SEGE CALL SEGE CALL INTR CALL INTR CALL PRNT CALL PRNT CALL SFEL CALL FFSL CALL LIST STOP DO 00 120 LB=1 IF(LB-EQ-1) CALL INIT CALL INTT	SLOPE(10), IDEN(80), CARI TI-T TI-T I-KI-MI-KJ-MJ-MIJ-LINE LOPE, IDEN, CAHO-TI-T ASYM) GO TO 100	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K+M+KS LOGICAL+1 SI LB=1 CALL INIT CALL INPT IF (TYPE-EQ- CALL CASE CALL SEGE CALL INTR CALL INTR CALL PRNT CALL PRNT CALL SLCE CALL SLCE CALL SLCE CALL SLCE CALL SLCE CALL ST STOP 00 00 120 LB=1 IF (LB-EQ-1) CALL INPT CALL INPT CALL INPT CALL INPT 10 CALL CASE	SLOPE(10), IDEN(80), CARI TI-T TI-T I-KI-MI-KJ-MJ-MIJ-LINE LOPE, IDEN, CAHO-TI-T ASYM) GO TO 100	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K-M-KS LOGICAL INIT CALL INPT IF (TYPE.EQ. CALL CASE CALL SEGE CALL INTR CALL PROFL CALL SLCE CALL SLCE CALL ROFL CALL SLCE CALL ROFL CALL QUCK CALL LIST STOP 00 00 120 LB=1 IF (LB-CQ-1) CALL INIT CALL INIT CALL INPT 10 CALL CASE CALL SEGE	SLOPE(10), IDEN(80), CARI TI-T TI-T I-KI-MI-KJ-MJ-MIJ-LINE LOPE, IDEN, CAHO-TI-T ASYM) GO TO 100	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K+M-KS LOGICAL+1 SI LB=1 CALL INIT CALL INPT IF (TYPE-EQ- CALL SEGE CALL SEGE CALL INTR CALL PRNT CALL PRNT CALL PRNT CALL ROFL CALL RFSL CALL LIST STOP 00 00 120 LB=1 IF (LB+CQ+1) CALL INTT CALL INTT CALL INTT 10 CALL CASE CALL INTR	SLOPE(10), IDEN(80), CARI TI-T TI-T I-KI-MI-KJ-MJ-MIJ-LINE LOPE, IDEN, CAHO-TI-T ASYM) GO TO 100	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K-M-KS LOGICAL*1 SI LB=1 CALL INIT CALL INPT IF (TYPE.EQ. CALL SEGE CALL INTR CALL SEGE CALL INTR CALL PRNT CALL PRNT CALL ROFL CALL SLCE CALL GUCK CALL LIST STOP 00 00 120 LB=1 IF (LB.EQ.1) CALL INPT CALL INPT CALL INPT CALL INPT 10 CALL CASE CALL SEGE CALL SEGE CALL GARY	SLOPE(10), IDEN(80), CARI TI-T TI-T I-KI-MI-KJ-MJ-MIJ-LINE LOPE, IDEN, CAHO-TI-T ASYM) GO TO 100	•KV•KRV	
COMMON /LA/ COMMON /LA/ COMMON /LS/ REAL K.M.KS LOGICAL*1 SI LB=1 CALL INIT CALL INPT IF (TYPE.EQ. CALL SEGE CALL SEGE CALL SEGE CALL PROFL CALL PROFL CALL SUCE CALL ROFL CALL GUCK CALL LIST STOP DO 120 LB=1 IF (LB.EQ.1) CALL INIT CALL INTT CALL INTT CALL LASE CALL SEGE CALL	SLOPE(10), IDEN(80), CARI TI-T TI-T I-KI-MI-KJ-MJ-MIJ-LINE LOPE, IDEN, CAHO-TI-T ASYM) GO TO 100	•KV•KRV	
COMMON /LA/ COMMON /LS/ REAL K+M+KS LOGICAL+1 SI LB=1 CALL INIT CALL INPT IF ITYPE-EQ. CALL SEGE CALL INTR CALL QARY CALL PRNT CALL ROFL CALL SUCE CALL FSL CALL QUCK CALL LIST STOP 0 00 120 LB=1 IF (LB-EQ-1) CALL INPT CALL INPT CALL INPT 10 CALL CASE CALL SEGE CALL GARY CALL GARY CALL GARY	SLOPE(10), IDEN(80), CARI TI-T TI-T I-KI-MI-KJ-MJ-MIJ-LINE LOPE, IDEN, CAHO-TI-T ASYM) GO TO 100	•KV•KRV	

```
120 CONTINUE
                                                                        5700
     CALL SLCE
                                                                        5800
     CALL RFSL
                                                                        5900
     DG 130 LB=4.8
                                                                        6000
     CALL COPY
                                                                        6100
  130 CONTINUE
                                                                        6200
     CALL BLCK
                                                                        6300
     CALL QK3D
                                                                        6400
     CALL LIST
                                                                        6500
     STOP
                                                                        6600
     END
                                                                        6700
6800
     SUBROUTINE INIT
                                                                        6900
C------
                                                                        7000
C INITIALIZE VARIABLES
                                                                        7100
                                                                        7200
     COMMON /RA/X(10) .Z(10) .TH(10) .H(10) .K(10) .R(10) .B(10) .M(10) .
                                                                        7300
    1 CURVE(10).DIR(10).X1(10).X2(10).Z1(10).Z2(10).T1(10).T2(10).
                                                                        7400
    2 KSI(10), ETA(10), RHO(10), XQ(11), ZQ(11), TQ(11,2), KRY(3), CL(2),
                                                                        7560
    3 KV(2) • V(10 • 3 • 3) • SL(3) • RHOSL(10 • 3) • CRVSL(10 • 3)
                                                                        7600
    4.VB(10.3.2.8).CRVB (10.8).XMAPAX(10).ZMAPAX(10).XS10(10).ZS10(10).
                                                                        7700
    5 BLB(8)
                                                                        7800
     COMMON /IA/ICASE(10).JCASE(30).NCASE(7).NP(3).NS(3).NB(8)
                                                                        7900
     COMMON /RS/KI-MI-KJ-MJ-MIJ-LINE-BLNK-RINF-ERRMAX-OPDN-FLLX-
                                                                        8000
    1 YUCL.ZERO,FORE,AFTE,TOPS.BOTS.OPUP.OGIV.PIO180.XJ.ZJ.XI.ZI.HI.
                                                                        8100
    Z RI-BI-THI-BJ-A1-A2-A3-BIJ-THJ-S-THIJ-SIJ-A1-HJ-RJ-A4-DX
                                                                        8200
    3.TYPE.ASYM.UPFO.UPAF.UPTO.UPBO.DNFO.DNAF.DNTO.DNBO
                                                                        8300
     COMMON /IS/IR-IW-IF-IP-IPRINT-NSEG-NSEGP1-ISTOP-USTOP-KSTOP-
                                                                        8400
    1 I.J.L.IPI.LMI.II.NPTS.IZ.ISL.LB.N.NMI.I3
                                                                        8500
     COMMON /LA/SLOPE(10).IDEN(80).CARD(80)
                                                                        8600
     COMMON /LS/T1.T
                                                                        6700
     REAL KOMOKSIOKIOMIOKJOMJOMIJOLINEOKVOKRV
                                                                        8880
     LOGICAL+1 SLOPE.IDEN.CARD.TI.T
                                                                        8900
1000 CONTINUE
                                                                        9000
     IF(LB.GT.1)GO TO 1100
                                                                        9100
     IPRINT=0
                                                                        9200
     1R=5
                                                                        9300
     I¥≠6
                                                                        9400
     IF=1
                                                                        9500
     IP=7
                                                                        9600
     ERRMAX=5.E-5
                                                                       9700
1100 NSEG=10
                                                                        9800
     PI0180=3.1415926/180.
                                                                       9900
     ISTOP=0
                                                                       10000
     JSTOP=0
                                                                       10100
     KSTOP=0
                                                                       10200
     RETURN
                                                                       10300
     END
                                                                       10400
10500
     SUBROUTINE INPT
                                                                       10600
C-----
                                                                       10700
C OBTAIN INPUT DATA
                                                                       10800
C-----
                                                                       10900
     COMMON /RA/X(10).Z(10).TH(10).H(10).K(10).R(10).B(10).H(10).
                                                                      11000
    1 CURVE(10).DIR(10):X1(10).X2(10).Z1(10).Z2(10).T1(10).T2(10).
                                                                       11100
    2 KSI(10) .ETA(10) .RHO(10) .XQ(11) .ZQ(11) .TQ(11.2) .KRY(3) .CL(2) .
                                                                      11200
```

```
16900
2006 FORMAT( ! INPUT CARDS !)
     WR1TE(IW.2007)
                                                                              17000
2007 FORMAT(* *.82(*-*))
                                                                              17100
2008 READ(IF.90001.END=2011)CARD
                                                                              17200
                                                                              17300
     WR1TE(IW,2009)CARD
2009 FORMAT(' | 1.80A1.11)
                                                                              17400
     60 TO 2008
                                                                              17500
2011 WRITE(IW.2007)
                                                                              17600
2013 CONTINUE
                                                                              17700
     IF (IF.NE.IP) REWIND IF
                                                                              17800
     READ (1R.2010.END=2020) IUEN
                                                                              17900
     READ (IR. INPUT. END=2040)
                                                                              18000
2019 WR1TE(IW.9050)
                                                                              18100
     WRITE(IW+2111)L0.8L0(L8) .
                                                                              18200
2111 FORMAT(* INPUT DATA: *.12.2x.A4)
                                                                              18300
     WRITE ([W.2]]2)
                                                                              18400
2112 FORMATITO IT. 6X. T. AT. IIX. ZT. IIX. THT. IIX. HT. IIX. KT. IIX.
                                                                              18500
    1 * R**11X** B**11X**RMG*}
                                                                              18600
     WR1TE (Iw.2113)
                                                                              18700
2113 FORMAT(***.117X.*CURVE DIR SLOPE*)
                                                                              18800
     WRITE(IW.2114)
                                                                              18900
2114 FORMAT(*. *+132(*-*))
                                                                              19000
     WRITE(IW+2115)(I+X(I)+Z(I)+TH(I)+H(I)+K(I)+R(I)+B(I)+RHO(I)+
                                                                              19100
    1CURVE(I) .DIR(I) .SLOPE(I) .
                                                                              19200
    2 I=1.10)
                                                                              19300
2115 FORMAT(10(* *,12,1P0E13.5,11x,2(1x,44),4x,41/))
                                                                              19400
     IF (TYPE.EQ.ASYM.AND.LB.LT.3)GO TO 2023
                                                                              19500
                                                                              19600
     DO 2118 L=1.3
     WRITE (IW+2116) SL(L)
                                                                              19700
2116 FORMAT(*0*.44. 3x.*x-SL*. 9x.*Z-SL*. 9x.*T-SL*. 8x.*RHO-S*/
                                                                              19800
    1 * **54(*-*)
                                                                              19900
     WRITE(1W-2117)(I-{V{I-J-L}-J=1-3}-RHOSL(I-L)-I=1-10)
                                                                              20000
2117 FORMAT(10(* *.12.1P4E13.5/))
                                                                              20100
2118 CONTINUE
                                                                              20200
     WRITE (IW.2021)
                                                                              20300
2021 FORMAT( 101.6x. * XMAPAX 1.7x. * ZMAPAX 1.8x. * XSID 1.9x. * ZSID 1
                                                                              20400
    1/1 1.54(1-1))
                                                                              20500
     WRITE(IW+2022)(I+XMAPAX(1)+ZMAPAX(I)+XSID(I)+ZSID(I)+I=1+10)
                                                                              20600
2022 FORMAT(10(* *.12.1P4E13.5/))
                                                                              20700
2023 WRITE(IM.INPUT1)
                                                                              20800
     WRITE (IW. 9050)
                                                                              20900
     GO TO 3000
                                                                              21000
2020 WRITE(IW-2030)
                                                                              21100
2030 FORMAT( COND CODE 2020: MISSING ID CARD)
                                                                              21200
     WRITE (IW-2035)
                                                                              21300
2035 FORMAT (*VEXECUTION HALTED*)
                                                                              21400
     STOP 2020
                                                                              21500
2040 WRITE (IW. 2050) IR
                                                                              21600
2050 FORMAT(* COND CODE 2040: EOF ON UNIT IR=*.12)
                                                                              21700
     WRITE (IW+2035)
                                                                              21800
     STOP 2040 . . .
                                                                              21900
3000 RETURN
                                                                              22000
     ENTRY INPT1
                                                                              22100
     IF (IF.NE.IP) IR=IF
                                                                              22200
     READ(IR.INPUT.END=2040)
                                                                              22300
     GO TO 2019
                                                                              22400
```

	RMAT(80A1) RMAT(*1*)						2
EN							ž
C+++++		*******	********	*******		*******	***** 2
	BROUTINE CASE						2
T							2
	INE ICASE						
							4
	MMON /RA/X(10)						. 2
	URVE(10)+DIR(1 S1(10)+ETA(10)						
	A(S)*A(J0*3*3) 21(In)*E(V(In)					,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
	B(10.3.2.8).CF					101-7510	_
	LB(8)	1,10,00	******		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		2
	MMON /IA/ICASE	(10) JCAS	E (30) . NCASE	(7) .NP(3) .	NS (3) .A	(8(8)	
	MMON' /RS/K[+M]						ā
	UCL . ZERO . FORE						11. 2
2 8	1.81.TH1.BJ.A1	.AZ,A3.61	J.THJ.S.THI	J.SIJ.AI.H	J.RJ.A4	+0X .	
	YPE . ASYM . UPFO						
	MMON /IS/IR+I				P.JSTOF	KSTOP.	4
	•J•L•IP1•L#1•]						-
	MMON /LA/SLOPE	(10).IDEN	(80) + CARD (8	0)	•		
	MMON /LS/TI+T			-			
	AL K+M+KSI+KI			•KHV			
	GICAL*1 SLOPE			100			3
	MENSION E(10+7 Ulvalence (E(1			•			
		2).2 (1)					
1 2		3) •TH(1)					3
		.4).H (1)		•			
•		.5) .K (1)		5			7
5	• (E ()	+6) .R (1)	1			4	
6	• (E (	1.73.B (1)	1	* .		, .	4
3000 CO				• •			
	INE NO. SEGMEN	ITS NSEG		-			
	3010 J=1.10						
	10-J+1						
	ASE (1)=0	TO	2010	•			
	(CURVE (I) .NE+	SLNK) GU IL	3010			·	
3010 CC	EG=I-1		•				
	ITE (1W+3015) NS	EG.					
	RMAT( ONUMBER		TS NSFG =1.	131			
	URE CURVE='L1				H SEG		
	3040 I=1.NSE		ar ar a second	, ,		. 45	
	3020 J=1.2	-				• •	
	(CURVE (1) .EQ.	(RV (J) ) GO	TO 3040				
3020 CC							
	ITE (IW. 3030) I	CURVE (I)	KRY(1).KRY	(2)			
	RMATI COND CO				NOT = 1	TO *+A4+	
1.4							
	TOP=1						
3040 CC							
	(ISTOP.EQ.1)51						

```
L#O
                                                                            28100
      DO 3050 J=1,7
                                                                             28200
      IF (E (I+J) .LT.RINF)L=L+2**J
                                                                             28300
 3050 CONTINUE
                                                                             28400
      IF (SLOPE (I) .EQ.T) L=L+256
                                                                             28500
      IF (CURVE(I).EQ.KRV(2))L=L+512
                                                                             28600
      DO 3060 J=1.21
                                                                             28700
      IF (L.NE.JCASE(J)) GO TO 3060
                                                                             28800
      ICASE(I)=J
                                                                             28900
      GQ TO 3080
                                                                             29000
 3060 CONTINUE
                                                                             29100
      ISTOP=1
                                                                             29200
      WRITE (1w.3070) I
                                                                             29300
 3070 FORMAT( COND CODE 3070: INPUT DATA FOR SEGMENT +: 12. DOES NOT MA
                                                                            29400
     1TCH ANY OF PROGRAMMED OPTIONS*}
                                                                             29500
 3080 CONTINUE
                                                                             29600
      WRITE(IW+3090)(I+ICASE(I)+1=1+10)
                                                                             29700
 3090 FORMAT(10(/* ICASE(*+12+*) =*+13})
                                                                             29800
      IF(ISTOP.EQ.01GO TO 4000
                                                                             29900
      WRITE (IW-2035)
                                                                             30000
      STOP 3070
                                                                             30100
 4000 RETURN
                                                                             30200
 2035 FORHAT ( OEXECUTION HALTED )
                                                                             30300
      END
                                                                            30400
30500
      SUBROUTINE SEGE
                                                                             30600
Corrections
                                                                            30700
TO DETERIMINE SEGMENT EQUATION
                                                                            30800
                                                                            30900
      COMMON /RA/X(10)+Z(10)+TH(10)+H(10)+K(10)+R(10)+B(10)+H(10)+
                                                                            31000
     1 CURVE(10).DIR(10).X1(10).X2(10).Z1(10).Z2(10).T1(10).T2(10).
                                                                            31100
     2 KSI(10) .ETA(10) .RHO(10) .XQ(11) .ZQ(11) .TQ(11.2) .KRV(3) .CL(2) .
                                                                            31200
     3 KV(2) + V(10+3+3) + SL(3) + RHOSL(10+3) + CRVSL(10+3)
                                                                            31300
     4.VB(10.3.2.8).CRVB (10.8).XMAPAX(10).ZMAPAX(10).XSID(10).ZSID(10).
                                                                            31400
     5 BLB(8)
                                                                            31500
      COMMON / IA/ICASE(10) + JCASE(30) + NCASE(7) + NP(3) + NS(3) + NB(8)
                                                                            31600
      COMMON /RS/KI.MI.KJ.MJ.MIJ.LINE.BLNK.RINF.ERRMAX.OPDN.ELLX.
                                                                            31700
     1 YUCL-ZERO-FORE-AFTE-TOPS-BOTS-OPUP-OGIV-PIO180-XJ-ZJ-XI-ZI-HI-
                                                                            31800
     Z RI-BI-THI-BJ-A1-A2-A3-BIJ-THJ-S-THIJ-SIJ-AI-HJ-RJ-A4-DX
                                                                            31900
     3.TYPE.ASYM.UPFO.UPAF.UPTO.UPBO.DNFO.DNAF.DNTO.DNBO
                                                                            32000
      COMMON /IS/IR.IW.IF.IP.IPRINT.NSEG.NSEGP1.ISTOP.USTOP.KSTOP.
                                                                            32100
     1 I.J.L.IPI.LMI.II.NPTS.IZ.ISL.LB.N.NMI.I3
                                                                            32200
      COMMON /LA/SLOPE(10), IDEN(80), CARD(80)
                                                                            32300
      COMMON /LS/TI.T
                                                                            32400
      REAL K.M.KSI.KI.MI.KJ.MJ.MIJ.LINE.KV.KRV
                                                                            32500
      LOGICAL®1 SLOPE, IDEN, CARD, TI.T
                                                                            32600
 4000 CONTINUE
                                                                            32700
      XJ=0.
                                                                            32800
      ZJ=0.
                                                                            32900
      DO 4990 I=1.NSEG
                                                                            33000
      Il=ICASE(I)
                                                                            33100
      IF (1.EQ.1) GG TO 4007
                                                                            33200
      IF (ICASE (I-1).NE.5.AND.ICASE (I-1).NE.9) GO TO 4007
                                                                            33300
      DO 4006 J=1.7
                                                                            33400
      IF (11.NE.NCASE (J)) GO TO 4006
                                                                            33500
      WRITE(IW+4005) [CASE([]+ICASE(I-1)
                                                                            33600
```

4005	
	FORMATI! COND CODE 4005: CASE 1.12. MAY NOT BE USED AFTER CASE
	112)
	ISTOP=1 · · · · · · · · · · · · · · · · · · ·
	GO TO 4007
4006	CONTINUE
4007	CONTINUE OF STANDARD TO A CONTINUE OF STANDARD S
	XI=X(I) in the state of the sta
	ZI=Z(I)
	HI#H(I)
	KI=K(I)
•	RI=R(I)
	BI*B(I) () (**)
	TI=SLOPE(I) - THI=TH(I) vo
¢ .	
Ċ	
,	0.70(4070:4090:4115:4120:4150:4010:4020:4030:4150:4160:
	1 - 4170,4180,4190,4200,4300,4410,4420,4430,4460,4490
- 7	5)•[] 1
ICA	
	TH(1)=ATAN(H(1))/PIO180
	8(1)=Z1-M(1)+X1
	GO TO 4040
I CA	
4020	M(I)=TAN(TH(I)=PI0180)
	Z(1) = M(1) = (X(1) + XJ) + ZJ
	B(I)=Z(I)-M(I)+XI
	60 TO 404U
1CA	56=8
4030	M(I)=TAN(TH(I)*PIO180)
	X(I)=XJ+(ZI=ZJ)/M(I)
	B(I)=ZI-M(I)+x(I)
	GO TO 4040
CHE	
	GO TO 4040
	GO TO 4040 CK REDUNUANT INPUT DATA
	GO TO 4040 CK REDUNDANT INPUT DATA JSTOP=0 IF(XI=LT=RINF.AND.ABS(XI=X([}).GT=ERRMAX)JSTUP=1 IF(JSTOP=EQ-1)WRITE(IW+4050)JSTOP+I=ERRMAX
	GO TO 4040 CK REDUNDANT INPUT DATA  JSTOP=0 IF (XI-LT-RINF.AND.ABS(XI-X(I)).GT.ERRMAX)JSTUP=1 IF (JSTOP-E0.1)WRITE(IW-4050)JSTOP+I-ERRMAX IF (ZI-LT-RINF.AND.ABS(ZI-Z(I)).GT.ERRMAX)JSTOP=2
	GO TO 4040 CK REDUNDANT INPUT DATA JSTOP=0 IF(XI.LT.RINF.AND.ABS(XI-X(I)).GT.ERRMAX)JSTUP=1 IF(JSTOP.EQ.1)WRITE(IW.4050)JSTOP.I.ERRMAX IF(JSTOP.EQ.1)WRITE(IW.4050)JSTOP.I.ERRMAX)JSTOP=2 IF(JSTOP.EQ.2)WRITE(IW.4050)JSTOP.I.ERRMAX
	GO TO 4040 CK REDUNDANT INPUT DATA JSTOP=0 IF (XI=LT-RINF.AND.ABS(XI=X(1)).GT.ERRMAX)JSTUP=1 IF (JSTOP=E0.1)WRITE (IW,4050)JSTOP.I.ERRMAX IF (ZI=LT-RINF.AND.ABS(ZI=Z(1)).GT.ERRMAX)JSTOP=2 IF (JSTOP.E0.2)WRITE (IW,4050)JSTOP.I.ERRMAX IF (BI=LT-RINF.AND.ABS(BI=0(1)).GT.ERRMAX)JSTOP=3
	GO TO 4040 CK REDUNDANT INPUT DATA  JSTOP=0  IF (XI = LT = RINF = AND = ABS (XI = X(I)) = GT = ERRMAX) JSTUP=1  IF (JSTOP = EQ = 1) WRITE (IW + 050) JSTOP + I = ERRMAX  IF (ZI = LT = RINF = AND = ABS (ZI = Z(I)) = GT = ERRMAX) JSTOP = 2  IF (JSTOP = EQ = 2) WRITE (IW + 050) JSTOP + I = ERRMAX JSTOP = 3  IF (JSTOP = EQ = 3) WRITE (IW + 050) JSTOP + I = ERRMAX JSTOP = 3  IF (JSTOP = EQ = 3) WRITE (IW + 050) JSTOP + I = ERRMAX
	GO TO 4040 CK REDUNUANT INPUT DATA  JSTOP=0  IF (XI.LT.RINF.AND.ABS(XI-X(I)).GT.ERRMAX)JSTUP=1  IF (XI.LT.RINF.AND.ABS(XI-X(I)).GT.ERRMAX  IF (ZI.LT.RINF.AND.ABS(ZI-Z(I)).GT.ERRMAX)JSTOP=2  IF (JSTOP.EQ.2) WRITE (IW.4050)JSTOP.I.ERRMAX  IF (BI.LT.RINF.AND.ABS(BI-B(I)).GT.ERRMAX)JSTOP=3  IF (JSTOP.EQ.3) WRITE (IW.4050)JSTOP.I.ERRMAX  IF (THI.LT.RINF.AND.ABS(THI.TH(I)).GT.ERRMAX
*0 <b>*</b> 0	GO TO 4040  CK REDUNDANT INPUT DATA  JSTOP=0  IF (XI-LT-RINF.AND.ABS(XI-X(I)).GT.ERRMAX) JSTOP=1  IF (XISTOP.E0.1) WRITE (IW.4050) JSTOP.I.ERRMAX) JSTOP=2  IF (JSTOP.E0.2) WRITE (IW.4050) JSTOP.I.ERRMAX  IF (BI.LT.RINF.AND.ABS(UI-B(I)).GT.ERRMAX JSTOP=3  IF (JSTOP.E0.2) WRITE (IW.4050) JSTOP.I.ERRMAX  IF (BI.LT.RINF.AND.ABS(UI-B(I)).GT.ERRMAX) JSTOP=4  IF (JSTOP.E0.3) WRITE (IW.4050) JSTOP.I.ERRMAX JSTOP=4  IF (JSTOP.E0.4) WRITE (IW.4050) JSTOP.I.ERRMAX
*040 *:	GO TO 4040  CK REDUNDANT INPUT DATA  JSTOP=0  IF (XI=LT-RINF.AND.ABS(XI=X(I)).GT.ERRMAX)JSTUP=1  IF (JSTOP.EQ.1) WRITE (IW.4050) JSTOP.I.ERRMAX  IF (ZI=LT.RINF.AND.ABS(ZI=Z(I)).GT.ERRMAX)JSTOP=2  IF (JSTOP.EQ.2) WRITE (IW.4050) JSTOP.I.ERRMAX  IF (BI=LT.RINF.AND.ABS(BI=0(I)).GT.ERRMAX)JSTOP=3  IF (JSTOP.EQ.3) WRITE (IW.4050) JSTOP.I.ERRMAX  IF (TMI=LT.RINF.AND.ABS(TMI=TH(I)).GT.ERRMAX)JSTOP=4  IF (JSTOP.EQ.4) WRITE (IW.4050) JSTOP.I.ERRMAX  FORMATI* COND CODE 4050-*:II.** REDUNDANT INPUT DATA FOR SEGMENT
::: :050	GO TO 4040 CK REDUNUANT INPUT DATA  JSTOP=0  IF (XI.LT.RINF.AND.ABS(XI-X(I)).GT.ERRMAX)JSTUP#1  IF (XI.LT.RINF.AND.ABS(XI-X(I)).GT.ERRMAX)JSTUP#1  IF (ZI.LT.RINF.AND.ABS(ZI-Z(I)).GT.ERRMAX)JSTOP=2  IF (JSTOP.EQ.2)WRITE(IW.4050)JSTOP.I.ERRMAX  IF (BI.LT.RINF.AND.ABS(BI-B(I)).GT.ERRMAX)JSTOP=3  IF (JSTOP.EQ.3)WRITE(IW.4050)JSTOP.I.ERRMAX  IF (THI.LT.RINF.AND.ABS(INI-TH(I)).GT.ERRMAX)JSTOP=4  IF (JSTOP.EQ.4)WRITE(IW.4050)JSTOP.I.ERRMAX  IF (JSTOP.EQ.4)WRITE(IW.4050)JSTOP.I.ERRMAX  IF (JSTOP.EQ.4)WRITE(IW.4050)JSTOP.I.ERRMAX
72 1050	GO TO 4040  CK REDUNDANT INPUT DATA  JSTOP=0  IF (XI-LT-RINF-AND-ABS(XI-X(I))-GT-ERRMAX) JSTOP=1  IF (XI-LT-RINF-AND-ABS(XI-X(I))-GT-ERRMAX) JSTOP=2  IF (JSTOP-EG-2) WRITE (IW-4050) JSTOP-I-ERRMAX  IF (BI-LT-RINF-AND-ABS(BI-B(I))-GT-ERRMAX) JSTOP=3  IF (JSTOP-EG-3) WRITE (IW-4050) JSTOP-I-ERRMAX  IF (BI-LT-RINF-AND-ABS(BI-B(I))-GT-ERRMAX) JSTOP=3  IF (JSTOP-EG-3) WRITE (IW-4050) JSTOP-I-ERRMAX  IF (JSTOP-EG-4) WRITE (IW-4050) JSTOP-I-ERRMAX  IF (JSTOP-EG-4) WRITE (IW-4050) JSTOP-I-ERRMAX  IF (JSTOP-EG-4) WRITE (IW-4050) JSTOP-I-ERRMAX  FORMAT(* COND CODE 4050-*-II-*)*** REDUNDANT INPUT DATA FOR SEGMENT  I-12-*** NOT SELF CONSISTENT TO WITHIN *-1PEIS-7**  IF (JSTOP-NE-0) WRITE (IW-405S)**
:040 :::	GO TO 4040  CK REDUNDANT INPUT DATA  JSTOP=0  IF (XI=LT-RINF.AND.ABS(XI=X(I)).GT.ERRMAX) JSTOP=1  IF (XI=LT-RINF.AND.ABS(XI=X(I)).GT.ERRMAX) JSTOP=2  IF (JSTOP.EG.) WRITE (IW.4050) JSTOP-I.ERRMAX  IF (BI=LT-RINF.AND.ABS(II=U(I)).GT.ERRMAX) JSTOP=3  IF (JSTOP.EG.3) WRITE (IW.4050) JSTOP-I.ERRMAX  IF (BI=LT-RINF.AND.ABS(III=U(I)).GT.ERRMAX) JSTOP=4  IF (THI=LT-RINF.AND.ABS(III=TH(I)).GT.ERRMAX) JSTOP=4  IF (JSTOP.EG.4) WRITE (IW.4050) JSTOP-I.ERRMAX  FORMAT(* COND CODE 4050-**-II,**! REDUNDANT INPUT DATA FOR SEGMENT  I= (JSTOP.EG.4) WRITE (IW.4055)  FORMAT(* SUGGEST USER CHECK IS INPUT DATA*)
050 055	GO TO 4040 CK REDUNUANT INPUT DATA  JSTOP=0  IF (XI.LT.RINF.AND.ABS(XI-X(I)).GT.ERRMAX)JSTUP#1  IF (XI.LT.RINF.AND.ABS(XI-X(I)).GT.ERRMAX)JSTUP#1  IF (XI.LT.RINF.AND.ABS(XI-Z(I)).GT.ERRMAX)JSTOP=2  IF (JSTOP.EQ.2) WRITE (IW.4050)JSTOP.I.ERRMAX  IF (BI.LT.RINF.AND.ABS(BI-B(I)).GT.ERRMAX)JSTOP=3  IF (JSTOP.EQ.3) WRITE (IW.4050)JSTOP.I.ERRMAX  IF (THILT.RINF.AND.ABS(THI-TH(I)).GT.ERRMAX)JSTOP=4  IF (JSTOP.EQ.4) WRITE (IW.4050)JSTOP.I.ERRMAX  IF (JSTOP.EQ.4) WRITE (IW.4050)JSTOP.I.ERRMAX  IF (JSTOP.EQ.4) WRITE (IW.4050)JSTOP.I.ERRMAX  IF (JSTOP.EQ.4) WRITE (IW.4050)JSTOP.I.ERRMAX  IF (JSTOP.EQ.4) WRITE (IW.4055)  FORMAT (** COND.CODE 4050-**-II.** REDUNDANT INPUT DATA FOR SEGMENT  IS (JSTOP.NE.0) WRITE (IW.4055)  FORMAT (** SUGGEST USER CHECK IS INPUT DATA*)  GO TO 4150
7: 1050 1055	GO TO 4040 CK REDUNUANT INPUT DATA JSTOP=0 IF (XI.LT.RINF.AND.ABS(XI-X(I)).GT.ERRMAX)JSTOP=1 IF (XI.LT.RINF.AND.ABS(XI-X(I)).GT.ERRMAX)JSTOP=1 IF (JSTOP.EQ) WRITE (IW.4050)JSTOP.I.ERRMAX)JSTOP=2 IF (JSTOP.EQ) WRITE (IW.4050)JSTOP.I.ERRMAX IF (BI.LT.RINF.AND.ABS(BI-B(I)).GT.ERRMAX)JSTOP=3 IF (JSTOP.EQ) WRITE (IW.4050)JSTOP.I.ERRMAX IF (HI.LT.RINF.AND.ABS(THI-TH(I)).GT.ERRMAX IF (JSTOP.EQ) WRITE (IW.4050)JSTOP.I.ERRMAX IF (JSTOP.EQ) WRITE (IW.4050)JSTOP.I.ERRMAX IF (JSTOP.EQ) WRITE (IW.4050)JSTOP.I.ERRMAX IF (JSTOP.EQ) WRITE (IW.4050)JSTOP.I.ERRMAX IF (JSTOP.NO) WRITE (IW.4055) FORMAT(* SUGGEST USER CHECK IS INPUT DATA*) GO TO 4150 SE=1:
T: +050 +055	GO TO 4040  CK REDUNDANT INPUT DATA  JSTOP=0  IF (XI-LT-RINF.AND.ABS(XI-X(I)).GT.ERRMAX) JSTOP=1  IF (JSTOP.EQ.1) WRITE (IW.4050) JSTOP+1.ERRMAX) JSTOP=2  IF (JSTOP.EQ.2) WRITE (IW.4050) JSTOP-1.ERRMAX  IF (BI-LT-RINF.AND.ABS(UI-U(I)).GT.ERRMAX) JSTOP=3  IF (JSTOP.EQ.3) WRITE (IW.4050) JSTOP-1.ERRMAX  IF (BI-LT-RINF.AND.ABS(UI-U(I)).GT.ERRMAX) JSTOP=3  IF (JSTOP.EQ.3) WRITE (IW.4050) JSTOP-1.ERRMAX  IF (JSTOP.EQ.6) WRITE (IW.4050) JSTOP-1.ERRMAX  IF (JSTOP.EQ.6) WRITE (IW.4050) JSTOP-1.ERRMAX  IF (JSTOP.EQ.6) WRITE (IW.4050)  IF (JSTOP.NE.0) WRITE (IW.4055)  FORMAT(* COND CODE 4050-*.II.*!* REDUNDANT INPUT DATA FOR SEGMENT  IF (JSTOP.NE.0) WRITE (IW.4055)  FORMAT(* SUGGEST USER CHECK IS INPUT DATA*)  GO TO 4150  SEE1:  MJ=-(XI-XJ)/(ZI-ZJ)
T: 4050 4055 ICA	GO TO 4040  CK REDUNDANT INPUT DATA  JSTOP=0  IF (XI=LT-RINF.AND.ABS(XI=X(I)).GT.ERRMAX)JSTUP=1  IF (XISTOP.EO.)) WRITE (IW.4050) JSTOP.I.ERRMAX)  IF (ZI=LT-RINF.AND.ABS(ZI=Z(I)).GT.ERRMAX)JSTOP=2  IF (JSTOP.EO.2) WRITE (IW.4050) JSTOP.I.ERRMAX  IF (BI.LT-RINF.AND.ABS(HI=G(I)).GT.ERRMAX)JSTOP=3  IF (JSTOP.EO.3) WRITE (IW.4050) JSTOP.I.ERRMAX  IF (THI-LT-RINF.AND.ABS(THI-TH(I)).GT.ERRMAX)JSTOP=4  IF (JSTOP.EO.4) WRITE (IW.4050) JSTOP.I.ERRMAX  FORMAT(* COND CODE 4050-*-11.*): REDUNDANT INPUT DATA FOR SEGMENT  IS (JSTOP.NE.0) WRITE (IW.4055)  FORMAT(* SUGGEST USER CHECK IS INPUT DATA*)  GO TO 4150  SE=1:  MJ=(XI-XJ)/(ZI-ZJ)  BJ=ZI-ZJ-MJ*(XI+XJ)
4040 4050 4055 ICAS	GO TO 4040 CK REDUNUANT INPUT DATA JSTOP=0 IF (XI.LT.RINF.AND.ABS(XI-X(I)).GT.ERRMAX)JSTUP=1 IF (XI.LT.RINF.AND.ABS(XI-X(I)).GT.ERRMAX)JSTUP=1 IF (XISTOP.EQ.1) WRITE (IW.4050)JSTOP-I.ERRMAX JSTOP=2 IF (JSTOP.EQ.2) WRITE (IW.4050)JSTOP-I.ERRMAX IF (BI.LT.RINF.AND.ABS(BI-BHI)).GT.ERRMAX)JSTOP=3 IF (JSTOP.EQ.3) WRITE (IW.4050)JSTOP-I.ERRMAX IF (JSTOP.EQ.3) WRITE (IW.4050)JSTOP-I.ERRMAX IF (JSTOP.EQ.4) WRITE (IW.4050)JSTOP-I.ERRMAX IF (JSTOP.EQ.4) WRITE (IW.4050)JSTOP-I.ERRMAX IF (JSTOP.EQ.4) WRITE (IW.4050)JSTOP-I.ERRMAX IF (JSTOP.EQ.4) WRITE (IW.4050)JSTOP-I.ERRMAX FORMAT(** COND CODE 4050-'+11.**; REDUNDANT INPUT DATA FOR SEGMENT IF (JSTOP.NE.0) WRITE (IW.4055) FORMAT(** SUGGEST USER CHECK IS INPUT DATA*) GO TO 4150 SE=1: MJ==(XI-XJ)/(ZI-ZJ) BJ=ZI-ZJ-MJ+(XI+XJ) BJ=ZI-ZJ-MJ+(XI+XJ) BJ=ZI-ZJ-MJ+(XI+XJ)
+0+0 +050 +055 ICA:	GO TO 4040  CK REDUNDANT INPUT DATA  JSTOP=0  IF (XI.LT.RINF.AND.ABS(XI-X(I)).GT.ERRMAX)JSTOP=1  IF (JSTOP.EQ.1) WRITE (IW.4050)JSTOP+1.ERRMAX)JSTOP=2  IF (JSTOP.EQ.2) WRITE (IW.4050)JSTOP-1.ERRMAX  IF (BI.LT.RINF.AND.ABS(BI-B(I)).GT.ERRMAX)JSTOP=3  IF (JSTOP.EQ.3) WRITE (IW.4050)JSTOP+1.ERRMAX  IF (BI.LT.RINF.AND.ABS(BI-B(I)).GT.ERRMAX)JSTOP=3  IF (JSTOP.EQ.3) WRITE (IW.4050)JSTOP+1.ERRMAX  IF (JSTOP.EQ.3) WRITE (IW.4050)JSTOP+1.ERRMAX  IF (JSTOP.EQ.3) WRITE (IW.4050)JSTOP+1.ERRMAX  IF (JSTOP.EQ.3) WRITE (IW.4050)JSTOP+1.ERRMAX  IF (JSTOP.EQ.3) WRITE (IW.4055)  FORMAT(* COND CODE 4050-*11.*1* REDUNDANT INPUT DATA FOR SEGMENT  IF (JSTOP.NE.0) WRITE (IW.4055)  FORMAT(* SUGGEST USER CHECK IS INPUT DATA*)  GO TO 4150  GO TO 4150  BJ=2(3/J-MJ)/(ZI-ZJ)  BJ=2(3/J-MJ)/(ZI-BJ)/(1.+MJ**2)
+0+0 +050 +055 ICA:	GO TO 4040 CK REDUNUANT INPUT DATA JSTOP=0 IF (XI.LT.RINF.AND.ABS(XI-X(I)).GT.ERRMAX)JSTUP=1 IF (XI.LT.RINF.AND.ABS(XI-X(I)).GT.ERRMAX)JSTUP=1 IF (XISTOP.EQ.1) WRITE (IW.4050)JSTOP-I.ERRMAX JSTOP=2 IF (JSTOP.EQ.2) WRITE (IW.4050)JSTOP-I.ERRMAX IF (BI.LT.RINF.AND.ABS(BI-BHI)).GT.ERRMAX)JSTOP=3 IF (JSTOP.EQ.3) WRITE (IW.4050)JSTOP-I.ERRMAX IF (JSTOP.EQ.3) WRITE (IW.4050)JSTOP-I.ERRMAX IF (JSTOP.EQ.4) WRITE (IW.4050)JSTOP-I.ERRMAX IF (JSTOP.EQ.4) WRITE (IW.4050)JSTOP-I.ERRMAX IF (JSTOP.EQ.4) WRITE (IW.4050)JSTOP-I.ERRMAX IF (JSTOP.EQ.4) WRITE (IW.4050)JSTOP-I.ERRMAX FORMAT(** COND CODE 4050-'+11.**; REDUNDANT INPUT DATA FOR SEGMENT IF (JSTOP.NE.0) WRITE (IW.4055) FORMAT(** SUGGEST USER CHECK IS INPUT DATA*) GO TO 4150 SE=1: MJ==(XI-XJ)/(ZI-ZJ) BJ=ZI-ZJ-MJ+(XI+XJ) BJ=ZI-ZJ-MJ+(XI+XJ) BJ=ZI-ZJ-MJ+(XI+XJ)

# AEDC-1R-//-89

	MARG. AB.: 48	
	H(I)=A1+A2	39300
	K(I)=MJ+H(I)+BJ	39400
	IF(DIR(I).EQ.OPDN.AND.K(I).LT.ZI)G0 TO 4080	39500
	H(1)=A1=A2	
		39600
	K(I)=MJ*H(I)+BJ	39700
4080	H(I)=-(X(I)-H(I))/{Z(I)-K(I)}	39800
	TH(I)=ATAN(H(I))/PI0180	39900
	60 TO 4130	40000
C ICAS		
		40100
4090	IF(I.6T.1)60 TO 4110	40200
	WRITE(IM-4100)	40300
4100	FORMATI COND CODE 41001 FIRST SEGMENT MAY NOT BE SPECIFIED TANGEN	40400
	IT TO PREVIOUS SEGMENT!)	40500
•	WRITE (IW. 2035)	
		40600
	STOP 4100	40700
4110	HJ=-(X-IX)/(X-IX)	40800
	0J=ZJ-EZJ-HJ+(XI+XJ)	40900
	0.0≈0.0 E	41000
	MI=-1./TAN(PI0180=TH(I-1))	
		41100
	BI-ZJ-HI-XJ	41200
	H(I)=-(BJ-BI)/(HJ-HI)	41300
	K(I)=MJ=H(I)+0J	41400
	R(I)=SQRT((XI=H(I))==2+(ZI=K(I))==2;	41500
	TH(I)=-ATAN((XI-H(I))/(ZI-K(I)))/PIO180	41600
	90 TO 4130	
C ICAS		41700
		41800
4115		41900
	IF(DIR(I).NE.OPDN)S==1.	42000
	H(I)=XJ+RI+SIN(PIO180+THJ)+S	42100
	K(1)=ZJ-RI+COS(PIO180+THJ)+S	42200
	Z(I)=K(I)+S+SQRT(RI+=2-(X1-H(I))++2)	42300
	TH(I)=-ATAN((XI-H(I))/(Z(I)-K(I)))/P10180	42400
	60 TO 4130	42500
C ICAS		42600
4120	MJ=-1./TAN(P[0180*TH(]))	42700
	IX+LM-IX=LB	42800
	MIJ=-(XJ-XJ)/(ZI-ZJ)	
	BIJ=ZI+ZJ-MIJ+(XI+XJ)	42900
		43000
	BIJ=0.5*BIJ	43100
	(LIM-LB)/(UIM-LB)/(UIM-LB)	43200
	K(I)=MJ=H(I)+BJ	43300
	R(I)=SQRT((XI=H(I))==2+(ZI=K(I))==2)	43400
	90 TO 4130	43500
	K REDUNDANT DATA	
		43600
4130	KSTOP=0	43700
	IF(XI .LT.RINF.AND.ABS(XI -X (I)).GT.ERRMAX)KSTOP=1	43800
	IF (KSTOP.EQ.1) WRITE (IW++1+0) KSTOP.I.ERRMAX	43900
	IF (ZI .LT.RINF.AND.ABS(ZI -Z (I)).GT.ERRMAX)KSTOP=2	44000
	IF (KSTOP.EQ.2) WRITE (IW.4140) KSTOP.I.ERRMAX	44100
	TE (THE ! T DINE AND ADDITH THE TAX OF COMMISSIONS	
	IF (THI.LT.RINF.AND.ABS(THI-TH(I)).GT.ERRMAX)KSTOP=3	44200
	IF (KSTOP.EQ.3) WRITE (IW.4140) KSTOP.I.ERRMAX	44300
	IF(HI -LT-RINF-AND-ABS(HI -H (I))-GT-ERRMAX)KSTOP=4	44400
	IF(KSTOP.EQ.4)WRITE(IW.4140)KSTOP.I.ERRMAX	44500
	IF(KI .LT.RINF.AND.ABS(KI -K (I)).GT.ERRMAX)KSTOP=5	44600
	IF (KSTOP.EQ.5) WRITE ([W.+140) KSTOP.I.ERRMAX	44700
	IF(RI .LT.RINF.AND.ABS(RI -R (I)).GT.ERRMAX)KSTOP=6	
	TO THE THE THIRD THE TREE TO STATE OF THE TREE TREE TO STATE OF THE TREE TREE TREE TREE TREE TREE TREE	44800

IF(KSTOP.EQ.6)WRITE(IW.+140)KSTOP.I.ERRMAX	
40 FORMATE COND CODE 4140-1-11-11 REDUNDANT INPUT DATA FOR SEGME	NT .
1.12. NOT SELF CONSISTENT TO WITHIN 1.1PE15.7)	
TELUPPOR AND ALLIANDER LAW AREE.	
IF(KSTOP,NE.0)WRITE(IW.+055) GO TO 4150 150 CONTINUE XJ=X(I)	
SO TO VIEW	
SO CONTINUE	
AU-X(I)	
ZJ=Σ(1)	
THJ=TH(I)	
GO TO 4990	
(CASE=10	
60 S=1.	
IF(018(1),F0,0PUP)S=1.	
M(1) = X Taseptes (N(PIO18. #THT)	
WITH-TI-CADIACOCIOTOLOGOTHII	
00 TO 4130	
00 10 413V	
CASE II	
70 S=1.	
IF (DIR(I).EQ.OPUP)S=+1.	
F(KSIOP*NE.0] WRITE(IW**USS)   GO TO 4150	
TH([)=-ATAN((XI-H1)/(Z(I)-KI))/PIO180	
GO TO 4130	
CASE=12	
TEINIBUTY EN ENDE NO DIGITY EN AFTERN TO ALOR	
AT TURNING ENGINE OUR OUR TIP SENSOR TE TOV TO TECH	
WAITE LEVEL CODE ALGE DID MINE DE EVENED ANGELLO DE CALCE	
BE LOWWILL COND CODE #182: DIN MORE RE ETIMEN OME ON WELS	
IURE SEGMENT'+13)	
85 FORMAT(* COND CODE 4185: DIR MUST BE EITHER **FORE** OR **AFTE 10RE SEGMENT*-13) WRITE([w.2035) 510P 4185	
STOP 4185	
86 CONTINUE	
IF (OIR (I) -FQ-FORE) S==1.	
1111=MIAS@SORT(RI@#2=(71-KI)@#2)	
TH(1) = ATAM/(K(1) - MI) ///I - K(1) ///I O) R6	
CO PO A130	
GO TO 4130  CASE=13  PO S=1.  IF(DIR(I).EQ.OPUP)S=-1.  X(I)=HI-SPRI*SIN(PID180*THI)  Z(I)=KI-SPRI*COS(PID180*THI)  GO TO 4130  CASE=14  DO B(I)=ZI=XI*TAN(PID180*THI)  GO TO 4040	
WRITE(IW-2035) STOP 4185 86 CONTINUE IF(DIR(I)-EQ.FORE)S=-1. X(I)=HIS-SSGRT(RI**2-(ZI-KI)**2) TH(I)=-ATAN((X(I)-HI)/(ZI-KI))/PIO180 GO TO 4130 CASE=13 90 S=1. IF(DIR(I)-FO.ORUP)S==1.	
90 S=1.	
IF(DIR(I).EQ.OPUP)S==1.	
Z(I)=KI+S=RI=COS(PIO180=THI)	
GO TO ♦130	
CASE=14	
ON BILLETT-KISTAM(DIGIBOSTHI)	
CO TO ADAD	
00 B(I)=ZI=XI=TAN(PIO180=THI) GO TO 4040 CASE=15	
CASE=15 00 Z(I)=XI=TAN(PIO180*THI)+BI	
JO Z(I) WXL TAN(PIOLBOTHL) FUL	
GO TO 4040	
CASE=16	
90 X(I)=(ZI-BI)/TAN(PIO180*TMI)	
60 10 4040	
CASE+17	
PADETIF	
In it. Judic (11) -XIII - INILIA (LITATON - INILI) -B(I)) -FIRMWY) 60 10 4120	
JSTOP=5	-
GO TO 4040  ICASE=16  POO X(I)=(ZI-BI}/TAN(PIO180*THI) GO TO 4040  ICASE=17  PLO IF (ABS(Z(I)-X(I)*TAN(PIO180*TH(I))*B(I))*LT*ERRMAX)GO TO 4150  JSTOP=5 WRITE(IW*4050)JSTOP*I*EHRMAX GO TO 4150  ICASE=18	
	*
GO TO 4150	

Table 8. Continued

```
4420 IF(ABS((X(I)-H(I))++2+(Z(I)-K(I))++2-R(I)++2).LT.ERRMAX)GO TO 4150
      WRITE(IW.4140)KSTOP.I.EHRMAX
                                                                              50700
      GO TO 4150
                                                                              50800
C ICASE=19
                                                                              50900
4430 S=0.
                                                                              51000
      IF (DIR(I) .EQ.UPFO) 5-+1.
                                                                              51100
      IF (DIR(I).EQ.UPAF)S=+1.
                                                                              51200
      IF (DIR(1) .EQ.UPTO) S=-1.
                                                                              51300
      IF (DIR(I) .EQ.UPBO) S=+1.
                                                                              51400
      IF (DIR(I) .EQ.ONFO) 5=-1.
                                                                              51500
      IF (DIR(I).EQ.DNAF)S=-1.
                                                                              51600
      IF (DIR(I).Eq.DNTO)S==1.
                                                                              51700
      1F (DIR(1).EQ.DNBO) S=-1.
                                                                              51800
      IF (S.NE.O.) GO TO 4440
                                                                              51900
      WRITE(IW-4431)1
                                                                              52000
 4431 FORMAT( COND CODE 4431: DIR(I) EQUAL TO ILLEGAL VALUE FOR CASE 19
                                                                              52100
     1 FOR SEGMENT .. 13)
                                                                              52200
      WRITE(1W+2035)
                                                                               52300
      STOP 4431
                                                                               52400
4440 IF (I.NE.NSEG) GO TO 4450
                                                                              52500
      WRITE (1W-4441)1
                                                                               52600
 4441 FORMAT( COND CODE 4441: ICASE=19 MAY NOT BE USED FOR LAST SEGMENT
                                                                              52700
                                                                              52800
    1 * + [3]
      WRITE (IW-2035)
                                                                              52900
      STOP 4441
                                                                              53000
 4450 H(I)=XJ-S=R1 =SIN(PIO180=THJ)
                                                                              53100
      K(1)=ZJ+S+RI +COS(P10180+THJ)
                                                                              53200
      IF(DIR(I).EQ.UPFO)DIR(I)=FORE
                                                                              53300
      IF(DIR(I).EQ.UPAF)DIR(I)=AFTE
                                                                              53400
      IF(DIR(I).EQ.UPTO)DIR(I)=TOPS
                                                                              53500
      1F(DIR(1).EQ.UPBO)DIR(1)=80TS
                                                                              53600
      IF (DIR (I) .EQ. DNFO) DIR (I) =FORE
                                                                              53700
      IF(DIR(I).EQ.DNAF)DIR(I)=AFTE
                                                                              53800
      IF (DIR(I).EQ.DNTO)DIR(I)=TOPS
                                                                              53900
      IF (DIR(I).EQ.ONBO)DIR(I)=BOTS
                                                                              54000
      ICASE (1) -5
                                                                              54100
      GO -10 -4130
                                                                              54200
C ICASE=20
                                                                              54300
4460 IH1=1-1
                                                                              54400
      A1=SQRT((XI=H(IM1))+>2+(Z1=K(IM1))+=2)
                                                                               54500
      A2=ARCOS (R (IM1) /A1) /P10180
                                                                               54600
      A3=ATAN((Z1-K(IM1))/(X1-H(IM1)))/P10180
                                                                               54700
                                                                               54800
      S=0.
      IF (DIR(I) .EQ. TOPS) S=+1.
                                                                               54900
      IF (DIR(I).Eq.BOTS)S=-1.
                                                                               55000
                                                                               55100
      IF(S.NE.O.) GO TO 4470
      WRITE(IW+4461)I
                                                                               55200
 4461 FORMAT ( COND CODE 4461: DIR MUST BE EITHER TOPS OR BOTS FOR SEGME
                                                                              55300
     INT*+13)
                                                                               55400
      WRITE(IW.2035);
                                                                               55500
      STOP 4461 -----
                                                                              55600
                                                                               55700
 4470 IF(I.NE.1)GC TO 4480
                                                                               55800
      WRITE(IW+4471)
 4471 FORMAT(* COND CODE 4471: ICASE=20 MAY NOT BE USED FOR FIRST SEGMEN
                                                                              55900
                                                                               56000
    17")
```

	WRITE(IW+2035)
4494	STOP 4471
4400	TH([]=A3-S*(90,-A2)
	M(I)=TAN(PI0180+TH(I))
	B([)=Z[-M([)=X[ GO TO 4040
C ICA	** · · · · · · · · · · · · · · · · · ·
- 10A	SEEL HE LLOO TO ARGO
4770	IF(I.NE.))GO TO 4500 WRITE(IW.4491)
4491	FORMAT(* COND CODE 4491: ICASE=2) MAY NOT HE USED WITH FIRST SEGEM
	LOWWRITE COUR CORE 44ATA TCWRE-SI WAT WAL BE AREA ATTU LIKRI REGEN
	write (Iw•2035)
	STOP 4491
.4500	IF(CURVE(I-1).EQ.OGIV)60 TO 4540
CITM	-OGIVE
	IM1=I+1
•	MJ STAN(PIG)80-TH(IN))
	A]=MJ=X(I)+B(IMI) Smal
	Sm+1.
••	S#*1. IF(Al.GT.Z(I))S#=1.
	BJ=8([M1)*5*R(I)/COS(PIU180*TH([M1))
	A1=1.+MJ==2
-	A2=2.*((I) J-Z(I)) **
	A3=X([) 002+(BJ-Z([))+02-R([)+02
	A4+A2/2./A1
	A5=A3/A1
	A5=A4**2~A5
	IF(A5.6E.0.)GO TO 4515
	IF(ABS(AS).GE.ERRMAX)GO TO 4510
	IF(ABS(A5).GE.ERRMAX)GO TO 4510 WRITE(IW.4501)I
4501	FORMAT (+ COND CODE 4501: SEGMENT+:13++ LOCUS OF CIRCLE CENTERS DO
	(ES NOT INTERSECT!)
	WRITE(IW-2035)
	STOP 4501
4510	AS=0.
4515	AS=SQRT (A5)
	A1=-A4+A5
	A2=-A4-A5
	LB-LM-14=EA
	LB+LM=SA=AA
	IF(A1.6T.A2)GO TO 4520
	H(I)=A1 K(I)=A3
	GO TO 4530 .
4520	H(I) = 2
	K(1)=46
4530	TH(I)=-ATAN((X(I)+H(I))/(Z(I)+K(I)))/PIO180
	GO TO ◆130
	/E-OGIVE
4540	IM1=I-1
_	IF(DIR(IM1).EQ.OPDN.OR.UIR(IM1).EQ.OPUP)GO TO 4550
	441444444444444444444444444444444444444
	FORMAT( * COND CODE 4542: SEGMENT . 13. DIR NOT EQUAL TO OPUP OR OP
:	lOn+)
	WRITE(IW-2035)
	STOP 4542

```
4550 IF(DIR(I).EQ.OPUP.OR.DIR(I).EQ.OPDN)GO TO 4560
                                                                               61700
      IM1=I
                                                                               61800
      GO TO 4541
                                                                               61900
4560 S=+1.
                                                                               62000
      IF(DIR(IM1).EQ.DIR(I))S==1.
                                                                               62100
      IF(Z(I).EQ.K(IM1))GO TO 4630
                                                                               62200
      MJ = (X(I) - H(IM1))/(Z(I) - K(IM1))
                                                                               62300
      BJ=X(I) **2+Z(I) **2-H(IH1) **2-K(IH1) **2+R(IH1) *(R(IH1) *2.*S*R(I))
                                                                               62400
      BJ=BJ+.5/(Z(I)-K(IM1))
                                                                               62500
      A1=1+HJ==2
                                                                               62500
      ((1)X-UH+((1)X-UB))+.5=SA
                                                                               62700
      A3=X(1) **2+(BJ-Z(1)) **2-R(1) **2
                                                                               62800
      A4=.54A2/A1
                                                                               62900
      A5=A3/A1
                                                                               63000
      A5#A4##2-A5
                                                                               63100
      IF (A5.GE.O.) GO TO 4580
                                                                               63200
      IF (ABS (AS) .GE.ERRMAX) GO TO 4570
                                                                               63300
      WRITE(IW+4561)I
                                                                               63400
4561 FORMAT( * COND CODE 4561: SEGMENT +13. LOCUS OF CIRCLE CENTERS DO
                                                                               63500
    1ES NOT INTERSECT*)
                                                                               63600
     WRITE(IW-2035)
                                                                               63700
     STOP 4561
                                                                               63800
4570 A500.
                                                                               63900
4580 AS=SORT(AS)
                                                                               64000
     A1=-44+A5
                                                                               64100
     A2=-44-45
                                                                               64200
     A3#MJ*A1+BJ
                                                                               64300
     L8+SA+LH=#A
                                                                               64400
4585 IF(DIR(IM1).EQ.DIR(I))GO TO 4622
                                                                               64500
     IF (DIR (IM1) .EQ.OPDN.AND.DIR (1) .EQ.OPUP) GO TO 4590
                                                                               64600
     IF (DIR (IM1) .EQ. OPUP. AND. DIR (I) .EQ. OPON) GO TO 4620
                                                                               64700
     STOP 4581
                                                                               64800
4590 IF(A4.GT.A3)GO TO 4610
                                                                               64900
4600 H(I)=A1
                                                                               65000
     K(1)=A3
                                                                              65100
     GO TO 4660
                                                                              65200
4610 H(I)=AZ .
                                                                              65300
     K(1)=A4
                                                                              65400
     GO TO 4660
                                                                              65500
4620 IF(A4.GT.A3)GD TO 4600
                                                                              65600
     GO TO 4610
                                                                              65700
4630 Al =(X(I) **2+Z(I) **2-(H(IM1) **2+K(IM1) **2)+R(IM1) *(R(IM1) *2.*
                                                                              65800
    1 S*R(I)))*.5/(X(I)-H(IM1))
                                                                              65900
     A2=R(1)==2=(A) -x(1))++2
                                                                              66000
     IF (A2.GT.0.) GD TO 4650
                                                                              66100
     IF (ABS (A2) .GE.ERRMAX) GO TO 4640
                                                                              66200
     WRITE(IW+4561)]
                                                                              66300
     STOP 4562
                                                                              66400
4640 A2=0.
                                                                              66500
4650 A2=SQRT(A2)
                                                                              66600
     A3=Z(I)+A2
                                                                              66700
     A4=Z(I)-A2
                                                                              66800
     A2=A1
                                                                              66900
     GO TO 4585
                                                                              67000
4622 IF(DIR(IM1).EQ.OPDN.AND.DIR(I).EQ.OPDN)GO TO 4620
                                                                              67100
     IF(DIR(IM1).E0.0PUP.AND.DIR(I).E0.0PUP)G0 TO 4590
                                                                              67200
```

	(4 <u>7</u> 0)	
	STOP 4623	6730
4660	TH(I)==ATAN((X(I)=H(I))/(Z(I)=K(I)))/PI0180	6740
4000	GO TO 4130	6750
4000	CONTINUE	6760
4770		
	IF (JSTOP.NE.O) WRITE (IW-2035)	6770
	IF (KSTOP.NE.O) WRITE (IW-2035)	6780
	IF(ISTOP.NE.O)WRITE(IW-2035)	6790
	IF (JSTOP.NE.0) STOP 4050	6800
.*	IF(KSTOP.NE.0)STOP 4140	6810
	IF(ISTOP.NE.0)STOP 4005	6820
	RETURN	6830
2035	FORMAT( OEXECUTION HALTED!)	6840
	END	6850
C++++		6860
_	SUBROUTINE INTR	6870
^		6880
•	PUTE SEGMENT INTERSECTIONS FOR ICASE=5.9	6890
-		
,		6900
	COMMON /RA/X(10)+Z(10)+TH(10)+H(10)+K(10)+R(10)+B(10)+H(10)+	6910
	1 CURVE(10)+DIR(10)+X1(10)+X2(10)+Z1(10)+Z2(10)+T1(10)+T2(10)+	6920
	2"KS[(10)+ETA(10)+RH0(10)+XQ(11)+ZQ(11)+TQ(11+2)+KRV(3)+CL(2)+	6930
	3 KV(2),V(10,3,3),SL(3),R-OSL(10,3),CRVSL(10,3)	6940
	4.VB(10.3.2.8).CRVB (10.8).XMAPAX(10).ZMAPAX(10).XSID(10).ZSID(10).	6950
9	5 BLB(6)	6960
	COMMON / 1A/ICASE(10) + JCASE(3D) + NCASE(7) + NP(3) + NS(3) + NB(B)	6970
		6980
	1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,X1,ZI,HI,	6990
	2 RI-BI-THI-BJ-A1-A2-A3-UIJ-THJ-S-THIJ-SIJ-AI-MJ-RJ-A4-DX	7000
	3.TYPE.ASYM.UPFO.UPAF.UPTO.UPBO.DNFO.DNAF.DNTO.DNBO	7010
•	COMMON /[S/IR.]W.IF.IP.IPRINT.NSEG.NSEGPI.ISTOP.JSTOP.KSTOP.	
٠,		7020
	1 I+J+L+1P1+LM1+11+MPT5+12+15L+LB+N+MH1+13	7030
	COMMON /LA/SLOPE(10), IDEN(80), CARD(80)	70+0
	COMMON /LS/TI+T	7050
	REAL KOMOKSIOKIOMIOKJOMJOMIJOLINEOKVOKRV	7060
	LOGICAL®1 SLOPE.IDEN.CARD.TI.T	7070
5000	CONTINUE	7080
	DO 5320 I=1,NSEG	7090
	IPI=I+1	7100
	IF(ICASE(I).NE.5.AND.ICASE(I).NE.9)GO TO 5320	7110
	IF(I.NE.NSEG)GO TO 5020	7120
	WRITE(IW-5010)	7130
5010	FORMAT( COND CODE 5010: EQUATION IS INSUFFICIENT FOR LAST SEGMENT	7140
	14)	- 7150
	WRITE(IW.2035)	
		7160
E A > A	STOP 5010	7170
JVE0	CONTINUE	7180
	IF (ICASE (I) .EQ.5.AND.CURVE(IP1) .EQ.QGIV) GO TO 5040	7190
	IF(ICASE(I).EQ.5.AND.CURVE(IP1).EQ.LINE)GO TO 5250	7200
	IF(ICASE(I).EQ.9.AND.CURVE(IP1).EQ.OGIV)GO TO 5260	7210
	IF(ICASE(I).EQ.9.AND.CURVE(IP1).EQ.LINE)GO TO 5240	7220
	WRITE(IW-5025)	7230
5025	FORMAT(+ COND CODE 50301 OOPS: PROGRAMMER ERROR.+)	7240
	WRITE(IW+2035)	7250
5030	STOP 5030	7260
	VE-OGIVE INTERSECTION	7270
5040	CONTINUE	7280

```
THIJ=ATAN2(K([P])-K([)+H([P])-H([))/P[0]80
                                                                                72900
      SIJ=SQRT((H(IP1)-H(I))++2+(K(IP1)-K(I))++2)
                                                                                73000
      AI=
                (R(1) ++2+51J++2-R(IP1) ++2)/(2.+R(1) +51J)
                                                                                73100
      IF (ABS(AI).GT.1.) AI=SIGN(1..AI)
                                                                                73200
      AI=ARCOS(AI)/PI0160
                                                                                73300
      XI=H(I)+R(I)+COS((THIJ+AI)+PIO180)
                                                                                73400
      X(I)=H(I)+R(I)+COS((THIJ-AI)+PIO180)
                                                                                73500
      IF(ABS(X(I)-XI) .LE.ERRMAX)GO TO 5090
                                                                                73600
      IF(I.LT.NSEG.AND.SLOPE(I+1).EQ.T)GO TO 5070
                                                                                73700
      IF(DIR(I).EQ.FORE)GO TO 5060
                                                                                73800
      IF(DIR(I).EQ.AFTE)GO TO 5080
                                                                                73900
      WRITE (IW. 5050) I
                                                                                74000
 5050 FORMAT( COND CODE 5050: DIR MUST BE EITHER **FORE** OR **AFTE** F
                                                                                74100
     10R SEGMENT +.12)
                                                                                74200
      WRITE (IW.2035)
                                                                                74300
      STOP 5050
                                                                                74400
 5060 IF(xI.GE.X(1))GO TO 5070
                                                                                74500
 5065 X(I)=XI
                                                                                74600
      Z(I) = K(I) + R(I) + SIN(PIO180 + (THIJ + AI))
                                                                                74700
      GO TO 5230
                                                                                74800
 5070 Z(I) = K(1) + R(I) + SIN(PIO180+(THIJ-AI))
                                                                                74900
      GO TO 5230
                                                                                75000
 5080 IF(XI.GT.X(1))GO TO 5065
                                                                                75100
      60 10 5070
                                                                                75200
 5090 ZI=K(I)+R(I)+SIN(PIO180+(THIJ+AI))
                                                                                75300
      Z(I)=K(I)+R(I)+SIN(PIO180+(THIJ-AI))
                                                                                75400
      IF(ABS(Z(1)-ZI).LE.ERRMAX)GO TO 5230
                                                                                75500
      IF(DIR(1).EQ.TOPS)G0 TO 5200
                                                                                75600
      IF(DIR(I).Eq.BOTS)G0 TO 5220
                                                                                75700
      WRITE (IW,5100) I
                                                                                75800
 5100 FORMAT( COND CODE 5100: DIR MUST BE EITHER **TOPS** OR **BOTS** F
                                                                               75900
     10R SEGMENT '.12)
                                                                                76000
      WRITE(1w,2035)
                                                                                76100
      STOP 5100
                                                                                76200
 5200 IF(Z1.GT.Z([))GO TO 5210
                                                                                76300
 5205 X(I)=H(I)+R(I)+COS(PIO180+(THIJ-AI))
                                                                                76400
      GO TO 5230
                                                                                76500
 5210 Z(I)=ZI [wf
                                                                                76600
      X(I) = H(I) + R(I) + COS(PIO180 + {THIJ+AI})
                                                                                76700
      GO TO 5230
                                                                                76800
 5220 IF(ZI.GT.Z(1))60 TO 5205
                                                                                76900
      GO TO 5210
                                                                               77000
 5230 TH(I)=-ATAN((X(I)-H(I))/(Z(I)-K(I)))/PI0180
                                                                               77100
      GO TO 5320
                                                                               77200
C LINE-LINE INTERSECTION
                                                                               77300
 5240 M(I )=TAN(PIO180+TH(I ))
                                                                               77400
      M(IP1)=TAN(P10180+TH(IP1))
                                                                               77500
      X(I) = -\{0(IP1) + 0(I)\}/(M(IP1) + M(I)\}
                                                                               77600
      Z(I) = (M(I) + B(IP1) - M(IP1) + B(I)) / (M(I) + M(IP1))
                                                                               77700
      GO TO 5320
                                                                               77800
C OGIVE-LINE INTERSECTION
                                                                               77900
 5250 CONTINUE
                                                                               18000
      M(IP1)=TAN(P10180=TH(IP1))
                                                                               78100
      MI=M(IP1)
                                                                               78200
      BI=B(IP1)
                                                                               78300
      (I)H=LH
                                                                               78400
```

		7.41				
	KJ=K(I)	10.1 10.1				
	RJ#R(I)	**				•
	GO TO 5270					
LIN	-081VE INTER	SECTION				
5260	CONTINUE					
	M(I)=TAN(PIO	180*TH(I}}				
	MI=M(I)					
	BI=B(I).					
	HJ#H(IP1)					
	KJ=K(IP1)					
	RJ=R(IP1)					
	GO TO 5270		· .			
2270	A1=1.+MI++2					
3 <b>4</b> 1 V		N = 14 1				
-	AZ=MI*(BI-KJ					
	18) •S**UH=EA		~2			
	A4=A2++2-A1+					
	IF (A4.GT.O.)					
	IF (ABS (A4) .L		10 5272		•	
	WRITE (IW.527	1) I		_		
		D CODE 5271	1 SQRT(<0)	FOR SEGME	NT',13,' N	O INTERSECTIO
1	N. *)					
	WRITE (IW-203	5} ••				
	STOP 5271					
5272	A4=0.					*
5273	A4#SQRT (A4)		•			
•	\(+A+SA-)=IX	<b>4</b> }				•
	4A-SA-)=(I)X		•		. *	* 1 · · · · · · · · · · · · · · · · · ·
	IF (ABS (X(I)-		AXIGO TO 5	300		
	IF (I.LT.NSE6				١	
	IF (DIR(I) .EQ			00 10 3301	•	
	IF (DIR (I) .EQ			•		
	WRITE (IW.528		2210			
- 200	MUTICITATIO	0.000C E390	. Ded wiet	00 517455		OR MAFTE . F
			. DIK MOZI	DE CTIME	· ·· PURE· ·	UK TIAFILTE
ı	OR SEGMENT.					
	WRITE (IM+203	5)				
	STOP 5280					•
	IF(XI.GT.X()	))60 TO.530	D	*		
5295	X(I)=XI					•
5300	Z(1)=MI+X(1)	+8I				
	GO TO 5320					**
5310	IF(XI.GT.X()	1160 TO 529	5			•
	GO TO 5300					
5320	CONTINUE	•				
•	RETURN		• •			
2035	FORMAT ( * OEXE	CUTION HALT	ED+1			-
	END					•
	SUBROUTINE Q					
	2004001146 A		·			*
	UP FINAL ARR			• • •		•
3E (	OL LIMME WAN	ATS PUR WULL	CA MODEL			
		<del></del>				H ( 1 A )
	COMMON /RA/X	(10) +2 (10) +	IN (10) (N(1)	UJ 9K (1U) 9F	(fTn) *A(10)	+M(10)+
	CURVE (10)+0					
	KSI(10).ETA					) •CF(5) •
	KV(2).V(10.					
4	·, VB(10·3·2·8	}.CRVB (10.	B) +XHAPAX (	10) +ZMAPA)	((10).XSID(	10) • ZS10(10) •

```
5 BLB (8)
                                                                           84100
      COMMON /IA/ICASE(10).JCASE(30).NCASE(7).NP(3).NS(3).NB(8)
                                                                           84200
      COMMON /RS/KI-MI-KJ-MJ-MIJ-LINE-BLNK-RINF-ERRMAX-OPDN-ELLX-
                                                                           84300
     1 YUCL.ZERO.FORE.AFTE.TOPS.BOTS.OPUP.OGIV.PIO180.XJ.ZJ.XI.ZI.MI.
                                                                           84400
     X G. AA. LA. LA. LIZ. LIHT. Z. LHT. LIB. EA. SA. LA. LB. INT. IB. IR.
                                                                           84500
     3. TYPE: ASYM: UPFO: UPAF: UPTO: UPBO: DNFO: DNAF: DNTO: DNBO
                                                                           84600
      COMMON /IS/IR-IW-IF-IP-IPRINT-NSEG-NSEGP1-ISTOP-JSTOP-KSTOP-
                                                                           84700
     1 I.J.L.IP1.LM1.I1.NPT5.I2.ISL.LB.N.NM1.I3
                                                                           84800
      COMMON /LA/SLOPE(10).IDEN(80).CARD(80)
                                                                           84900
      COMMON /LS/TI.T
                                                                           85000
      REAL KOMOKSIOKIOMIOKJOMJOMIJOLINEOKVOKRV
                                                                           85100
      LOGICAL*1 SLOPE.IDEN.CARD.TI.T
                                                                           85200
C TQ(I+1) IS AT BEGINNING OF SEGMENT I
                                                                           85300
C TO(I.2) IS AT END SEGMENT I
                                                                           85400
C XQ(I) AND ZQ(I) ARE AT BEGINNING OF SEGMENT I
                                                                           85500
 6000 CONTINUE
                                                                           85600
      XQ(1)=0.
                                                                           85700
      ZQ(1)=0.
                                                                           65800
      DO 6010 I=1.NSEG
                                                                           85900
      IF (CURVE(I).EQ.KRV(2)) CURVE(I) =ELLX
                                                                           85000
      XQ(I+1)=X(I)
                                                                           86100
      ZQ(I+1)=Z(I)
                                                                           86200
 6010 CONTINUE
                                                                           86300
      DO 6030 I=1.NSEG
                                                                           86400
      IP1=1+1
                                                                           86500
      IF(CURVE(I).NE.LINE)GO TO 6020
                                                                           86600
      TQ(1+1)=ATAN((ZQ(IP1)=ZQ(I))/(XQ(IP1)=XQ(I)))/PIO180
                                                                           86700
      TQ([.2)=TQ([.1)
                                                                           86800
      GO TO 6030
                                                                           86900
 6020 IF(ZQ(1).EQ.K(1))GO TO 6025
                                                                           87000
      TQ([+1)=-ATAN((XQ([)-H(1))/(ZQ([)-K(1)))/P[Q]80
                                                                           87100
      60 TO 6024
                                                                           87200
 6025 TO([,1)=90.
                                                                           87300
6024 TQ([,2)=-ATAN((XQ([P1)+H([))/(ZQ([P1)-K([)))/PIO180
                                                                           87400
6030 CONTINUE
                                                                           87500
      NSEGP1=NSEG+1
                                                                           87600
C SET ALL RIGHT ANGLES (90 DEG) TO "IBM RIGHT ANGLES" (89.999 DEG)
                                                                           87700
      DO 6040 I=1.NSEGP1
                                                                           87800
      DO: 6040 J=1.2
                                                                           B7900
      IF(TQ(I+J).GT.RINF)GD TO 6040
                                                                           88000
      IF( ABS(TQ(1,J)).LE.89.999)GO TO 6040
                                                                           88100
      ((L+1)@T.999.98)ND12#(L+1)@T
                                                                           88200
6040 CONTINUE
                                                                           88300
      RETURN
                                                                           88400
                                                                           88500
88600
      SUBROUTINE PRNT
                                                                           88700
C-----
                                                                           88800
C PRINT RESULTS
                                                                           88900
C-----
                                                                           89000
     COMMON /RA/X(10)+Z(10)+TH(10)+H(10)+K(10)+R(10)+B(10)+H(10)+
                                                                           89100
     1 CURVE(10) +DIR(10) +X1(10) +X2(10) +Z1(10) +Z2(10) +T1(10) +T2(10) +
                                                                           89200
    2 KSI(10).ETA(10).RHO(10).XQ(11).ZQ(11).TQ(11.2).KRV(3).CL(2).
                                                                           89300
    3 KV(2)+V(10+3+3)+SL(3)+RHOSL(10+3)+CRVSL(10+3)
                                                                           89400
    4.VB(10.3.2.8).CRV8 (10.8).XMAPAX(10).ZMAPAX(10).XSID(10).ZSID(10).
                                                                           89500
    5 BL8(8)
                                                                           89600
```

· · · · · · · · · · · · · · · · · · ·	and the strategic CEA in the second of the s	
	インサータを開発する。	
	COMMON /IA/ICASE(10).JCASE(30).NCASE(7).NP(3).NS(3).NB(8)	89700
-	COMMON /RS/KI.MI.KJ.MJ.MIJ.LINE.BLNK.RINF.ERRNAX.OPDN.ELLX.	89800
	1 YUCL.ZERO.FORE.AFTE.TOPS.BOTS.OPUP.OGIV.PIO180.XJ.ZJ.XI.ZI.HI.	89900
	XI.48. THI.48. AR. AR. AR. AR. AR. AR. AR. AR. AR. AR	90000
	3+TYPE+ASYM+UPFO+UPAF+UPTO+UPBO+DNFO+DNAF+DNTO+DNBO	90100
	COMMON /IS/IR.IW.IF.IP.IPRINT.NSEG.NSEGP1.ISTOP.JSTOP.KSTOP.	90200
	1 [I+J+L+IP1+LM1+II+NPTS+IZ+ISL+LB+N+NM1+I3	90300
	COMMON /LA/SLOPE(10):IDEN(80);CARD(80)	90400
	COMMON /LS/TI-T	90500
	REAL K.M.KSI.KI.MI.KJ.MJ.HIJ.LINE.KV.KRV	90600
	LOGICAL®1 SLOPE.IDEN.CARD.TI.T	90700
7000	CONTINUE	90800
	UNITED ATT TAKEN	90900
7005	FORMATIANCE MENT FOR BOTH COORDINATE AND CLORECTE	91000
1003	######################################	
7010	WRITE LIBORULU)	91100
1010	PURMATE	91200
	00 1020 1=1111	91300
	write(Iw+90008)I+XQ(I}+ZQ(I}+TQ(I+2)+TQ(I+1)	91400
7020		91500
	write (Iw. 7025)	91600
7025	FORMAT (*OSEGMENT EQUATIONS:*)	91700
	WRITE(IW-7030)	91800
7030	FORMAT( '0"- 9x+*X*+13x+*H*+13X+*K*+13X+*R*+13X+*M*+13X+*T*+	91900
	2 13x, 181,/1 1,100(1-1))	92000
	DO 7040 I=1-10	92100
	WRITE([W.9000B)[.X([].H([].K([].R([).H([).TH(]).B([).	92200
7040	WRITE LIMPYOUDDITAKIJOMILIORIJORIJORIJORIJORIJORIJORIJORIJORIJORI	
,040	CONTINUE TO A SECOND	92300 -
	RETURN	92400
90008	FORMAT(***,12,1P9E14,6)	92500
	END A SECTION OF THE	92600
C++++		92700
	SUBROUTINE ROFL	92800
		92900
C INS	ERT ROUNDS AND FILLETS	93000
C		93100
•	COMMON /RA/X(10)+Z(10)+TH(10)+H(10)+K(10)+R(10)+B(10)+H(10)+	93200
	CURVE(10) +OIR(10) +X1(10) +X2(10) +Z1(10) +Z2(10) +T1(10) +T2(10) +	93300
	2 KSI(10) +ETA(10) +RHO(10) +XO(11) +ZO(11) +TO(11,2) +KRV(3) +CL(2) +	
		93400
	3 KY(Z).V(10.3.3).SL(3).RHOSL(10.3).CRYSL(10.3)	93500
	4.VB(10.3.2.8).CRV8 (10.8).XMAPAX(10).ZMAPAX(10).XSID(10).ZSID(10).	93600
l.	5 BLB(8)	93700
	COMMON /IA/ICASE(10):JCASE(30):NCASE(7):NP(3):NS(3):NB(8)	93800
		93900
	l-YUCL+ZERO+FORE+AFTE+TOPS+BOTS+OPUP+OGIV+P10180+XJ+ZJ+XI+ZI+HI+	94000
	XG.4A.UH.IA.UH.IA.UIZ.UIMT.2.UHT.UIB.EA.SA.IA.UB.IHT.IB.IR S	94100
	3. TYPE - ASYMALIPED - LIPAF - LIPTO - LIPRO - DNEO - DNAF - DN TO - DNRO	94200
•	COMMON /IS/IR-IH-IF-IP-IPRINT-NSEG-NSEGP1-ISTOP-KSTOP-	94300
	1 I.J.L. IPI.L. II. NPTS. 12. ISL. LB. N. NM1. 13	94400
	COMMON /IS/IR-IW-IF-1P-IPRINT-NSEG-NSEGP1.ISTOP-JSTOP-KSTOP.  1 I-J-L-IP1-LM1.II.nPTS-IZ-ISL-LB-N-NM1-I3 COMMON /LA/SLOPE(10).IDEN(80).CARD(80) COMMON /LS/II.T	
	CAMAN A EAT T	94500
	COMMON /LS/TI-T	94600
	REAL KOMAKSIAKIAMIAKJAMJAMIJALINEAKVAKRV	94700
	COSTCAC-1 SCOPE-IDENICARD-11-1	94800
6100	CONTINUE	94900
	ISKIP=0	95000
	DO 6800 1=1.9	95100
	IP1=I+1	95200

```
IF (RMO(1).GT.RINF)GO TO 6800
                                                                                95300
      ISKIP=1
                                                                                95400
      IF (CURVE (I) .EQ.LINE.AND.CURVE (IP1) .EQ.LINE) GO TO 6110
                                                                               95500
      IF (CURVE(I).EQ.LINE.AND.CURVE(IP1).EQ.ELLX)GO TO 6200
                                                                               95600
      IF(CURVE(I).EQ.ELLX.AND.CURVE(IP1).EQ.LINE)GO TO 6300
                                                                               95700
      IF (CURVE(I).EQ.ELLX.AND.CURVE(IP1).EQ.ELLX)GO TO 6600
                                                                               95888
      WRITE ([w.6101) [
                                                                                95900
 6101 FORMAT( COND CODE 6101: PROGRAM OR USER ERROR. SEGMENT .. 13)
                                                                               96000
      WRITE (IN-2035)
                                                                                96100
      STOP 6101
                                                                                96200
C LINE-LINE FILLETTING/ROUNDING
                                                                                96300
 6110 CONTINUE
                                                                               96400
      IF (ABS(TQ(I.2)-TQ(IP1.1)).GT.ERRMAX)GO TO 6120
                                                                               96500
      WRITE (IW-6)11) I. IP1
                                                                               96600
 6111 FORMATI COND CODE 6111: FILLETS/ROUNDS MAY NOT BE INSERTED BETWEE
                                                                               96700
     INLINE SEGMENTS WITH SAME SLOPE SEGMENTS . . 214)
                                                                               96800
      WRITE (IW-2035)
                                                                               96900
      STOP 6111
                                                                               97000
 6120 S=1.
                                                                               97100
      IF (TO(IP1.1).6T.TO(I.2))5=-1.
                                                                               97200
      BI=B(I)-S-RHO(I)/COS(PI0180+TQ(I.2))
                                                                               97300
      BJ=B(IP1)-S=RHO(I)/COS(PI0180=TQ(IP1-1))
                                                                               97400
      M(1)=TAN(PIG180=TH(I))
                                                                               97500
      M(IP1)=TAN(P10180+TH(IP1))
                                                                               97600
      ETA(I) = (M(IP1) = 0 = M(I) = 0) / (M(IP1) = M(I))
                                                                               97700
      KSI(I)=-(BJ-BI)/(M(IP1)+M(I))
                                                                               97800
      X1(I) = KSI(I) - S + RHO(I) + SIN(PIO180 + TH(I))
                                                                               97900
      X2(I)=KSI(I)-S=RHO(I)=SIN(P10180=TH(IP1))
                                                                               98000
      21(1)=ETA(1)+S=RHO(1)=COS(P10180=TH(1 ))
                                                                               98100
      Z2(1)=ETA(1)+S=RHO(1)=COS(PIO180=TH(1P1))
                                                                               98200
      T1 (1) = TH(1)
                                                                               98300
      T2(1)=TH(1P1)
                                                                               98400
      GO TO 6800
                                                                               98500
C LINE-OGIVE FILLETTING/ROUNDING
                                                                               98600
6200 CONTINUE
                                                                               98700
C OGIVE-LINE FILLETTING/ROUNDING
                                                                               98800
6300 CONTINUE
                                                                               98900
 6201 S1=0.
                                                                               99000
      S2=0.
                                                                               99100
      S3=0.
                                                                               99200
      Ile!
                                                                               99300
      IF (CURVE (IP1) .EQ.ELLX) I1=IP1
                                                                               99400
      12=I
                                                                               99500
      IF (I1.EQ. I) 12=IP1
                                                                               99600
      IF (TQ(I.2)-TQ(IP1.1))6210.6205.6230
                                                                               99700
6205 WRITE (IW.6206) I, [P]
                                                                               99800
6206 FORMAT( COND CODE 6206: FILLETS/ROUNDS MAY NOT BE INSERTED BETWEE
                                                                               99900
     IN SEGMENTS WITH SAME SLOPE. SEGMENTS: 213)
                                                                              100000
      WRITE(IW.2035)
                                                                              100100
      STOP 6206
                                                                              100200
C TH(IP1)>TH(I)
                                                                              100300
6210 IF (K(I1)-20(IP1))6220,6215,6225
                                                                              100400
6215 WRITE (IW-6216) I. IP1
                                                                              100500
6216 FORMAT( COND CODE 6216: K = Z NOT A PROGRAMMED OPTION, SEGMENTS.,
                                                                              100600
    1 213)
                                                                              100700
      WRITE (14,2035)
                                                                              100800
```

	STOP 6216	100900
6220	S1=1.	101000
	S2=1.	101100
	\$3=1 <b>.</b>	101200
	60 TO 6250	101300
6225	S1=1.	101400
	\$2=-1.	101500
	\$3=1.	101600
	GO TO 6250	101700
6230	IF(K(I1)=Z0(IP1))6240,6235,6245	101800
	IP1) <th(i)< td=""><td>101900</td></th(i)<>	101900
6240	51=-1.	102000
	S2=-1.	102100
	53=-1.	102200
	GO TO 6250	102300
6235	WRITE(IW+6236)I+IP1	102400
6236	FORMAT( COND CODE 6236: K = Z NOT A PROGRAMMED OPTION, SEGMENTS	102500
	1 213)	102600
	write(lw, 2035)	102700
	STOP 6236	102800
6245	S1=-1.	102900
	S2=1.	103000
	\$3=-1.	103100
	GO TO 6250	103200
6250	CONTINUE	103300
	BI=B(I2)+S1=RHO(I)/COS(PIO180+TH(I2))	103400
	A1=1.+M(I2)++2	103500
	A2=2.+((BI-K(I1))+M(I2)-H(I1))	103600
	A3=H(I1)++2+(BI-K(I1)}-+2-(R(I1)+52+RHO(I))++2	103700
	A4=0.5 <b>4</b> AZ/A1	103800
	A5=A3/A1 '	103900
	A5=A4++2-A5	104000
	IF(A5.GE.O.)GO TO 6260	104100
	IF(ABS(A5).GT.ERRMAX)GO TO 6254	104200
	A5=0.	104300
	GO TO 6260	104400
	WRITE(IW,6255)I,IP1	104500
6255	FORMAT( COND CODE 6255: REAL SOLUTION TO FILLET/ROUND PROBLEM DO	104600
1	les not exist. Possible program error. Segments •,212)	104700
	WRITE(IW-2035)	104800
	STOP 6255	104900
6260	A5=SQRT (A5)	105000
	A1=-A4-A5	105100
	A2=H([2] •A1 •B1	105200
	A6=H([])+R(])+(A1-H(]))/SQRT((A1-H(]))+=2+(A2-K(]))+=2)	105300
	A7=(M(12)+(A2-B(12))+A1)/(1.+M(12)++2)	105400
	IF(X(I)-GE.AMIN1(A6+A7)-AND.X(I)-LE.AMAX1(A6+A7))GO TO 6265	105500
	A1=-A4+A5	105600
	A2=H(IZ) *A1+B1	105700
	A6=H([1])+R([1])+([1])/SQRT((A1-H([1]))+2+(A2-K([1))++2)	105800
	A7=(H([2)+(A2-8(12))+A1)/(1.+H(12)++2)	105900
	IF (X(1).GE.AMIN) (A6.A7).AND.X(1).LE.AMAX1(A6.A7))GO TO 6265	106000
	WRITE(IW-6261)I	106100
	FORMATI' COND CODE 6261: PROGRAM UNABLE TO CHOOSE BETWEEN TWO FILL	106200
	IET/ROUND SOLUTIONS, SEGMENT, 13)	106300
	WRITE(IW+2035)	106400

111300

111400

111500

111600

111700

111800

111900

112000

```
STOP 6261
                                                                             106500
 6265 KSI(I)=Al
                                                                             106600
      ETA(I)=A2
                                                                             106700
      IF(CURVE(I).EQ.ELLX)60 TO 6270
                                                                             106800
      X1(1)=A7
                                                                             106900
      X2(1)=A6
                                                                             107000
      Z2(1)=K(11)+R(11)+(ETA(1)-K(11))/SQRT((A1-H(11))++2+(A2-K(11))++2)
                                                                             107100
      Z1(I)=M(I2)+x1(I)+B(I2)
                                                                             107200
      T1(I)=TH(I)
                                                                             107300
      T2(I)=-ATAN((X2(I)-H(I1))/(Z2(I)-K(I1)))/PI0180
                                                                             107400
      GO TO 6800
                                                                             107500
 6270 X1(I)=A6
                                                                             107600
      X2(1)=A7
                                                                             107700
      Z2(1)=M(12)+X2(1)+B(12)
                                                                             107800
      Z1(I)=K(I1)+R(I1)+(ETA(I)-K(I1))/SQRT((A1-H(I1))++2+(A2-K(I1))++2)
                                                                             107900
      T1(I)=-ATAN((X1(I)-H(I1))/(Z1(I)-K(I1)))/PI0180
                                                                             108000
      T2(I)=TH(IP1)
                                                                             108100
      GO TO 6800
                                                                             108200
6600 CONTINUE
                                                                             108300
C OGIVE-OGIVE FILLETTING ROUNDING
                                                                             108400
      A1=TO(IP1.1)-TO(I.2)
                                                                             108500
      A2=Z(1)-K(1)
                                                                             108600
      A3=Z(I)-K(IP1)
                                                                             108700
      IF (A1.GT.ZERO.AND.A2.GT.ZERO.AND.A3.GT.ZERO) GO TO 6610
                                                                             108800
      IF (AL-LT-ZERO-AND-A2-GT-ZERO-AND-A3-GT-ZERO) GO TO 6615
                                                                             108900
      IF(A1.LT.ZERO.AND.A2.LT.ZERO.AND.A3.LT.ZERO)GO TO 6610
                                                                             109000
      IF(A1.GT.ZERO.AND.A2.LT.ZERO.AND.A3.LT.ZERO)GO TO 6615
                                                                             109100
      IF(A1.LT.ZERO.AND.A2.GT.ZERO.AND.A3.LT.ZERO)GO TO 6620
                                                                             109200
      IF(A1.GT.ZERO.AND.A2.GT.ZERO.AND.A3.LT.ZERO)GO TO 6625
                                                                             109300
      IF (Al.GT.ZERO.AND.AZ.LT.ZERO.AND.A3.GT.ZERO) GO TO 6620
                                                                             109400
      IF (Al.LT.ZERO.AND.AZ.LT.ZERO.AND.AJ.GT.ZERO) GO TO 6625
                                                                             109500
      WRITE(IW-6601) I, IP1, A1, A2, A3
                                                                             109600
6601 FORMAT ( COND CODE 6601: SEGMENTS + 213 . . A1 . A2 . A3 MAY NOT BE ZERO
                                                                            109700
    1. Al.AZ,A3=+,1P3E15.7)
                                                                             109800
      WRITE (IW. 2035)
                                                                             109900
      STOP 6601
                                                                             110000
6610 51=1.
                                                                             110100
      SZ=1.
                                                                             110200
      GO TO 6630
                                                                             110300
6615 S1=-1.
                                                                             110400
      52=-1.
                                                                             110500
      GO TO 6630
                                                                            110600
6620 S1=-1.
                                                                            110700
     S2=1.
                                                                            110800
      GO TO 6630
                                                                            110900
6625 51=1.
                                                                            111000
     52-1.
                                                                            111100
      GO TO 6630
                                                                            111200
```

BI=(H(IP1)++2-H(I)++2+K(IP1)++2-K(I)++2-RJ++2)/2./(K(IP1)+

6630 CONTINUE

1 K(I))

A1=1.+MI++2

RI=R(I)+S1+RHO(I)

RJ=R(IP1)+S2+RH0(I)

IF(K(I).EQ.K(IP1))GO TO 6645

MI=-(H(IP1)-H(I))/(K(IP1)-K(I))

	AB-D ALIDE MITTINGE MITTI	112100
	A2=2.*((BI-K(I))*MI-H(I))	112100
	A3=H(I)++2+(BI-K(I))++2-RI++2	112200
	445*42/41	112300
	A5=A3/A1	112400
	A5=A4**2-A5	112500
	IF(AS.GE.O.)GO TO 6635	112600
	IF(ABS(AS).LE.ERRMAX)GO TO 6634	112700
	WRITE(IW,6631)I,IP1	112800
	FORMAT(* COND CODE 6631: REAL SOLUTION TO FILLET/ROUND PROBLEM DOE	112900
1	IS NOT EXIST. POSSIBLE PROGRAM ERROR. SEGMENTS*,2I3)	113000
	WRITE(IW+2035)	113100
	STOP 6631	113200
6634	A5=0.	113300
6635	A5=SQRT (A5)	113400
	A1=-A4+A5	113500
	A2=-A4-A5	113600
	A3=MI+A1+BI	113700
	A4=MI*AZ+BI	113800
6636	A5=SQRT((X(I)-A1)++2+(Z(I)-A3)++2)	113900
	A6=SQRT((X(I)-A2)+2+(Z(I)-A4)+2)	114000
	IF(A6.LT.A5)G0 TO 6640	114100
	KSI(I)=Al	114200
	ETA(I)=A3	114300
	GO TO 6650	114400
6640	KSI(I)=A2	114500
0040	ETA(I)=A4	114600
	GO TO 6650	114700
4445	Al=(H([P])++2=H([)++2=RJ*+2+R[++2}/2./(H([P])+H([))	114800
9043		114900
	A2=A1	115000
	A4=RI++2-(A1-H(I))++2	115100
	IF (A4.GE.0.) GO TO 6648	
	IF (ABS(A4),GT,ERRMAX)GO TO 6646	115200
	A4=0.	115300
00-0	A4=SQRT (A4)	115400
	A3=K(I) -A4	115500
	A4=K(I)+A4	115600
	60 10 6636	115700
	WRITE(IM+6647)I+IP1	115800
	FORMAT(* COND CODE 6647: FOR SEGMENTS*+213+ K(1) = K(1P1): REAL S	115900
	lolution to fillet/round problem does not exist. Possible program e	116000
i	2RROR.*)	116100
	WRITE(IW+2035)	116200
	STOP 6647	116300
6650	DO 6660 J=I.IP1	116400
	A1=SQRT((KS1(1)-H(J))+#2+(ETA(1)-K(J))##2)	116500
	A2=H(J) +R(J) + (KSI (I) -H(J) ) /A1	116600
	A3=K (J) +R (J) + (ETA (Ī) -K (J) ) /AĪ	116700
	0810191((J))/(L) N-EA) / 04-EA)	116800
	IF (J.EQ.1P1) GO TO 6655	116900
	X1(I)=A2	117000
	Z1(I)=A3	117100
	T1(I)=A6	117200
	GO TO 6660	117300
6655	1) SX	117400
	Z2(I)=A3	117500
	T2(I)=A4	117600

6660 CONTINUE	117700
6800 CONTINUE	117800
IF (ISKIP.EQ.0) RETURN	117900
C PRINT RESULTS OF FILLET/ROUND COMPUTATION	118000
WRITE(IW+6810)	118100
6810 FORMAT( OFILLET/ROUND EQUATIONS AND INTERSECTIONS: */	118200
1'0', 7x,' X ',10x,'KSI',10x,'ETA',10x,'RHO',10x,' X1',10x,' Z1',	
2 10X, T1',10X, X2',10X, Z2',10X, T2'/	118400
31 1,132(1-1))	118500
WRITE(IW-6815)(I-X(I)-KSI(I)-ETA(I)-RMO(I)-X1(I)-Z1(I)-T1(I)- 1 X2(I)-Z2(I)-T2(I)-I=1-10)	118600
6815 FORMAT(10(* *, I2, 1P10E13.5/))	118700
C INSERT FILLET/ROUND DATA INTO QUICK ARRAYS	118800
6900 CONTINUE	118900
J=0	119000 119100
DO 6920 I=1.10	119200
J=J=1	119300
IF(RHO(I).GT.RINF)GO TO 6920	119400
J=J+1	119500
NSEG=NSEG+1	119600
NSEGP1=NSEG+1	119700
IF(NSEG.LE.10)GO TO 6905	119800
WRITE(IW.6901)I	119900
6901 FORMAT( COND CODE 6901 INSERTION OF ROUND : 13, EXCEEDS DIMENS	IO 120000
INS (10) OF QUICK*)	120100
WRITE (IW. 2035)	120200
STOP 6901	120300
6905 L=11	120400
6910 L=L-1	120500
[H]=[-]	120600
IF(L.LE.J)60 TO 6915	120700
XQ (L) =XQ (LH])	120800
ZQ(L)=ZQ(LM1)	120900
TQ(L,1)=TQ(LM1,1) TQ(L,2)=TQ(LM1,2)	121000
CURVE (L) =CURVE (LM1)	121100
GO TO 6910	121200
6915 XQ(J)=X1(I)	121300 121400
ZQ(J)=Z1(I)	121500
TQ(J+1)=T1(1)	121600
TQ(J,2)=T2(1)	121700
TQ(J-1,2)=T1(I)	121800
CURVÉ (J+1) =ĈUŘÝE (J)	121900
XQ(J+1)=X2(I)	122000
Z9(J+1)=Z2(Î)	122100
TQ(J+1-1)=T2(I)	122200
CURVE (J) =ELLX	122300
6920 CONTINUE	122400
C PRINT RESULTS	122500
WRITE(IW-6930)	122600
6930 FORMAT(*OFINAL QUICK ARRAY!*)	122700
WRITE(IW,7010)	122800
00 6940 [=1,10	122900
6940 WRITE(IW,90008)[,XQ([),ZQ([),TQ([,2),TQ([,1)	123000
RETURN 2035 FORMAT (AGENERALITANI, MAI VEDA)	123100
2035 FORMAT(*0EXECUTION HALTED*)	123200

	FORMAT('0'.9 x.'X'.13x.'2'.12x.'TA'.12X.'TF'/' '.58('-'))	123300
		123400
	FORMAT( * '.12.1P9E14.6)	123500
	END	
Cassas	495050000000000000000000000000000000000	123600
	SUBROUTINE QUCK	123700
C		123600
Č SET	UP QUICK MODEL	123900
	0.00	124000
	COMMON /RA/X(10).Z(10).TH(10).H(10).K(10).R(10).B(10).H(10).	124100
	CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),	124200
2	KSI(10).ETA(10).RHO(10).XQ(11).ZQ(11).TQ(11.2).KRV(3).CL(2).	124300
-	KV(2) .V(10.3.3) .SL(3) .RHOSL(10.3) .CRVSL(10.3)	124400
,	.VB(10.3.2.8).CRVB (10.8).XMAPAX(10).ZMAPAX(10).XSID(10).ZSID(10).	124500
		124600
	8LB(8)	124700
	COMMON /IA/ICASE(10) .JCASE(30) .NCASE(7) .NP(3) .NS(3) .NB(8)	124800
_	COMMON /RS/KI.MI.KJ.MJ.MIJ.LINE.BLNK.RINF.ERRMAX.OPDN.ELLX.	124900
1	YUCL.ZERO.FORE.AFTE.TOPS.BOTS.OPUP.OGIV.PIO180.XJ.ZJ.XI.ZI.HI.	
2	RI.BI.THI.BJ.A1.A2.A3.BIJ.THJ.S.THIJ.SIJ.A1.HJ.RJ.A4.DX	125000
3	.TYPE.ASYM.UPFO.UPAF.UPTO.UPBO.DNFO.DNAF.DNTO.DNBO	125100
	COMMON /IS/IR.1W.IF.IP.IPRINT.NSEG.NSEGP1.ISTOP.JSTOP.KSTOP.	125200
	I,J,L,IP1,LM1,I1,NPT5,I2,ISL,LB,N,NM1,I3	125300
	COMMON /LA/SLOPE(10).IDEN(80).CARD(80)	125400
	COMMON /LS/TI.T	125500
	REAL K.M.KSI.KI.MI.KJ.MJ.MIJ.LINE.KV.KRV	125600
	LOGICAL-1 SLOPE.IDEN.CARD.TI.T	125700
	CONTINUE	125800
• • • •	IF(IF.NE.1P)REWIND IF	125900
C CROS	S SECTION MODELS	126000
	WRITE (IF. 90001) IDEN	126100
	IF(ISL.NE.0)GO TO 8001	126200
	WRITE(1F.90002)	126300
	WRITE(1F.90003)	126400
	GO TO 8002	126500
8001	WRITE(IF.21700)	126600
	CONTINUE	126700
OVVE	11=1	126800
	WRITE(IF,90004)]],II,XQ(1),XQ(NSEGP1)	126900
C 800v	LINE MODELS	127000
C OVUI		127100
	WRITE(IF.90005)YUCL	127200
	WRITE(IF.90006) II. KRY(I) KY(I)	127300
	WRITE (IF, 90007) ZERO. ZERO. XQ (NSEGP1). ZERO. ZERO. ZERO	127400
	WRITE(IF,90009)	127500
	00 8030 J=1+2	127600
	WRITE (IF • 90005) CL (J)	127700
	DO 8010 I=1.NSEG	
	IP1=I+1	127800
	11=1	127900
	IF (CURVE (I) .EQ.ELLX) I1=2	128000
	WRITE(IF,90006)I.CURVE(I).KV(I1)	128100
	GO TO (8006,8005)+11	128200
8005	CONTINUE	128300
	WRITE(IF.90007)XQ(I).ZQ(I).XQ(IP1).ZQ(IP1).TQ(I.1).TQ(I.2)	128400
	GO TO 8010	128500
8006	WRITE(IF,90014)XQ(I),ZQ(I),XQ(IP1)+ZQ(IP1)	128600
8010	CONT INUE	128700
	WRITE(IF,90009)	128800

Table 8. Continued

	DO 8020 I=1,NSEGP1	128900
	ZQ(I)=-ZQ(I)	129000
	TQ(I-1)=-TQ(I-1)	129100
	TQ(1,2)=-TQ(1,2)	129200
	CONTINUE	129300
9070	CONTINUE	129400
	WRITE(IF,90010)	129500
	IF(ISL.EQ.0)GO TO 8033 DO 8032 L=1.3	129600
	WRITE(IF,90005)SL(L)	129700
	I14NS(L)	129800
	DO 8031 I=1.I1	129900
	12=1	130000
	IF (CRVSL (I.L).EQ.ELLX) I2=2	130100 130200
	WRITE(IF.90006) I.CRVSL(I.L),KV(12)	130300
	WRITE(IF+90007) V(I+1+L)+V(I+2+L)+V(I+1+1+L)+V(I+1+2+L)+V(I+3+L)+	130400
	1 V(I+1+3+L)	130500
8031	CONTINUE	130600
	WRITE(IF.90009)	130700
803Z	CONTINUE	130800
	WRITE(IF.21900)	130900
8033	CONTINUE	131000
	WRITE(IF,90011)	131100
	IF (IPRINT.EQ.0) GO TO 8050	131200
	DX=.00001	131300
	NPTS=5	131400
C SEI	UP CARDS TO EXERCISE MODEL	131500
	11=1	131600
	IZ=Z	131700
	DO 8040 I=1,NSEG Al=xQ(I)+Dx	131800
	A2=XQ(I+1)=DX	131900
	A3=(A2-A1)/(NPTS-1)	132000
	A2=XQ(I+1)+DX	132100
	WRITE(1F.90012)11.12.41.42.43	132200 132300
8040	CONTINUE	132400
	IF(ISL-EQ.0)60 TO 8050	132500
	[1=2	132600
	12=2	132700
	I4=NP(1)	132800
	A3=2.	132900
	A4=-90.	133000
	A5=90.	133100
	A6=5.	133200
	DO 8044 L=1,3	133300
	13=NP (L)	133400
	00 8043 1=2,13	133500
	IF (L.EQ.1) GO TO 8042	133600
	1F(I.EQ.J)GO TO 8043	133700
	DO 8041 J=1, [4	133800
2041	IF(ABS(V(J+1+1)-V(I+1+L)).LE.ERRMAX)GO TO 8043 CONTINUE	133900
	A1=V(I+1+L)	134000
944E	A2=A1+1.	134100
	WRITE(IF.90012) [1.12.A1.A2.A3.A4.A5.A6	134200
8043	CONTINUE	134300
J		134400

0001 FORMAT(80A1) 0002 FORMAT(* 1', 1' 2') 0003 FORMAT(* 1', 1' 2') 1							
QUICK INPUT FORMATS							
OUICK INPUT FORMATS  OUICK INPUT FORMATS  OUO2 FORMAT(*1'*)*   2*)  1	8050 WRITE(	IF•90013)					
OUDIT FORMAT(80A1)  1	RETURN	•					
OUDIT FORMAT(80A1)  1	C						
0001 FORMAT(*  1/*  1 2*) 0002 FORMAT(*  1/*  1 2*) 1							
0003 FORMAT(' 1 i'' 1 2') 0003 FORMAT('LELL   ELLI PIEC   BLCL   SID   LSCP'/ 2							
0003 FORMAT(' 1 i'' 1 2') 0003 FORMAT('LELL   ELLI PIEC   BLCL   SID   LSCP'/ 2	90001 FORMAT	(BOA1)					
0003 FORMAT(!ELL 1ELLI PIEC SLCL SID LSCP!    'UELL 2ELLI PIEC SID TUCL USCP!   'NELL 2ELLI PIEC SID			211				
1				DIEC	PL CL	610	1.66847
0004 FORMAT(22,5x.2F10,5) 0005 FORMAT(3,1x,44,' PIEC ',44) 0006 FORMAT(3,1x,44,' PIEC ',44) 0006 FORMAT(3,1x,5,12,19,5,*,4'),F10,5) 0008 FORMAT(',12,199E14,6) 0009 FORMAT(',12,199E14,6) 0009 FORMAT(',12,199E14,6) 0019 FORMAT(',12,199E14,6) 0010 FORMAT(',12,199E14,6) 0010 FORMAT(',12,199E14,6) 0010 FORMAT(',12,199E14,6) 0010 FORMAT(',12,199E14,6) 0011 FORMAT(',12,199E14,6) 0011 FORMAT(',12,199E14,6) 0012 FORMAT(',12,13,5x,6F10,5) 0013 FORMAT(',12,13,5x,6F10,5) 0013 FORMAT(',12,13,5x,6F10,5) 0014 FORMAT(',12,13,5x,6F10,5) 0015 FORMAT(',12,13,5x,6F10,5) 0015 FORMAT(',12,13,5x,6F10,5) 0016 FORMAT(',12,13,5x,6F10,5) 0017 FORMAT(',12,13,5x,6F10,5) 0018 FORMAT(',12,13,5x,6F10,5) 0019 FORMAT(',12,13,5x,6F10,5) 0019 FORMAT(',12,13,5x,6F10,5) 0011 FORMAT(',13,13,5x,6F10,5) 0012 FORMAT(',13,13,5x,6F10,5) 0013 FORMAT(',13,13,5x,6F10,5) 0014 FORMAT(',13,13,5x,6F10,5) 0014 FORMAT(',13,13,5x,6F10,5) 0015 FORMAT(',13,13,5x,6F10,5) 0016 FORMAT(',13,13,5x,6F10,5) 0017 FORMAT(',13,13,5x,6F10,5) 0017 FORMAT(',13,13,5x,6F10,5) 0018 FORMAT(',13,13,5x,6F10,5) 0019 FORMAT(',13,13,5x,6F10,5) 0019 FORMAT(',13,13,5x,6F10,5) 0019 FORMAT(',13,13,5x,6F10,5) 0019 FORMAT(',13,13,5x,6F10,5) 0019 FORMAT(',13,13,5x,6F10,5) 0019 FORMAT(',10,13,13,5x,6F10,5) 0011 FORMAT(',10,13,13				_ = = = =			
0005 FORMAT(212.05.2F10.5) 0005 FORMAT(32.1x.A4,' PIEC ',A4) 0006 FORMAT(3710.5.2(F9.5.'x').F10.5) 0007 FORMAT(3710.5.2(F9.5.'x').F10.5) 0008 FORMAT('-1') 0010 FORMAT('-1') 0010 FORMAT('-1') 0110 FORMAT('YLCL YUCL'/ 2 'ZMAPAX YUCL'/ 3 'ZSID YUCL'/ 4 'YUSCP ZUCL'/ 5 'ZUSCP ZUCL'/ 6 'YLSCP ZUCL'/ 7 'YSID ZUCL'/ 7 'YSID ZUCL'/ 8 'ZLSCP ZLCL') 0011 FORMAT('') 0012 FORMAT('') 0013 FORMAT('') 0013 FORMAT(') 1 6 NONCIRCULAR CROSS SECTION '.21x/ 2'LELL IELLI PIEC BLCL SID LSCP '.21x/ 3'UELL ZELLI PIEC SID TUCL USCP '.21x/ 4'LBSL 3LINE PIEC BLSL LSLCP SLELL',11x/ 6'LSSL 4LINE PIEC BLSL LSLCP SLELL',11x/ 6'LSSL 4LINE PIEC SSL USLCP BUELL',11x/ 9'UTSL 6LINE PIEC SSL USLCP BUELL',11x/ 9'UTSL 6LINE PIEC USLCP TUSL UELL',21x/ 1 1 MAPAX') 1 1 MAPAX') 1 1 YLSL YUCL'.66x/ 1 'YLSL YUCL'.66x/ 2 C ZSSL ZSID'.66x/ A 'ZUSLCP ZUSL'.66x/ C "ZSSL ZSID'.66x/ END  SUBROUTINE LIST  LIST GUICK INPUT FILE  COMMON /RA/X(10),Z(10).TH(10).H(10).K(10).R(10).B(10).H(10),					210	TUCL	USCPI
0005 FORMAT(IA) 0006 FORMAT(IZ)IX,A4,* PIEC ',A4) 0007 FORMAT(IZ)IX,52(F9.5,*A*)*F10.5) 0008 FORMAT('*)IZ,1P9E14.6) 0009 FORMAT('*)IZ,1P9E14.6) 0010 FORMAT('*)IZ,1P9E14.6) 0010 FORMAT('YLCL YUCL'/ 1				. • ,			
0006 FORMAT(12:1x:AA,* PIEC *,AA) 0007 FORMAT(3:10:5,2(F9:5,*A*)*F10:5) 0008 FORMAT(* ',12:199E14:6) 0009 FORMAT(* ',12:199E14:6) 0010 FORMAT(* ',12:199E14:6) 0010 FORMAT(* ',12:199E14:6) 0010 FORMAT(* ',12:199E14:6) 1			. 10.2)				
0007 FORMAT(3F10.5-2(F9.5.*A*)*F10.5) 0008 FORMAT(**,12*,199E14.6) 0009 FORMAT(**,12*,199E14.6) 0010 FORMAT(**,12*,199E14.6) 0010 FORMAT(**,12*,109E14.6)  1							
0008 FORMAT(' ' 12,1P9E14.6) 0010 FORMAT('-1') 0010 FORMAT('-1') 1							
0009 FORMAT("YLCL YUCL"/  1				· ) • F10 • 5	5)		
1			9E14.6)				
YMAPAX YUCL'/   2							
2 'ZMAPAX YUCL'/ 3 'ZSID YUCL'/ 4 'YUSCP ZUCL'/ 5 'ZUSCP ZUCL'/ 6 'YLSCP ZUCL'/ 8 'ZLSCP ZLCL') 8 'ZLSCP ZLCL') 8011 FORMAT(*) 1012 FORMAT(*2,13,5%,6F10.5) 0013 FORMAT(*0,0) 0014 FORMAT(*0,0) 1700 FORMAT(*1,78%/ 1 1 6 NONCIRCULAR CROSS SECTION 2 'LELL IELLI PIEC BLCL SID LSCP ',21%/ 3 'UELL 2ELLI PIEC BLCL SID LSCP ',21%/ 4 'LBSL 3LINE PIEC BLSL LSLCP 5LELL',11%/ 6 'LSSL 4LINE PIEC BLSL LSLCP 5LELL',11%/ 6 'LSSL 4LINE PIEC SSL USLCP 8UELL',11%/ 7 'USSL 5LINE PIEC SSL USLCP 8UELL',11%/ 9 'UTSL 6LINE PIEC USLCP TUSL UELL',21%/ A '	900lo FORMAT	('YLÇL	YUCL	1			
2 'ZMAPAX YUCL'/ 3 'ZSID YUCL'/ 4 'YUSCP ZUCL'/ 5 'ZUSCP ZUCL'/ 6 'YLSCP ZUCL'/ 8 'ZLSCP ZLCL') 8011 FORMAT('') 0012 FORMAT(12,13,5%,6F10.5) 0013 FORMAT(2,13,5%,6F10.5) 0013 FORMAT(0 0') 0014 FORMAT(1 1',78%/ 1' 1 6 NONCIRCULAR CROSS SECTION '.21%/ 2'LELL 1ELLI PIEC BLCL SID LSCP '.21%/ 4'LBSL 3LINE PIEC BLSL LSLCP 5LELL',11%/ 6'LSSL 4LINE PIEC BLSL LSLCP 5LELL',11%/ 6'LSSL 4LINE PIEC SID TUCL USCP '.21%/ 7'USSL 5LINE PIEC SSL USLCP 8UELL',11%/ 9'UTSL 6LINE PIEC USLCP SSL LELL',21%/ 7'USSL 5LINE PIEC USLCP TUSL UELL',21%/ A '	1	'YMAPAX	YUCL!	/			
3	2	"ZMAPAX					
'YUSCP ZUCL'/   5	3	·ZSID	YUCL	/			
\$ 'ZUSCP ZUCL'/ 6 'YLSCP ZUCL'/ 7 'YSTD ZUCL'/ 8 'ZLSCP ZLCL') 0011 FORMAT(' ') 0012 FORMAT(!Z-;3,5x,6F10.5) 0013 FORMAT(3F10.5,F9.5) 1700 FORMAT(* 1.78x/ 1' 1 6 NONCIRCULAR CROSS SECTION		TYUSCP	ZUCL *	/			
6	5	ZUSCP	ZUCL *	1			
7							
8							
0011 FORMAT((* ') 0012 FORMAT((2)13,5x,6F10.5) 0013 FORMAT((0)) 0014 FORMAT(3F10.5,F9.5) 1700 FORMAT(* 1*,78%/ 1* 1 6 NONCIRCULAR CROSS SECTION			=				
0012 FORMAT(12-13,5x.6F10.5) 0013 FORMAT(* 0*) 0013 FORMAT(* 0*) 0014 FORMAT(* 0*) 1700 FORMAT(* 1*,78x/  1* 1 6 NONCIRCULAR CROSS SECTION 2*LELL 1ELLI PIEC BLCL SID LSCP '.21x/ 3*UELL 2ELLI PIEC SID TUCL USCP '.21x/ 4*LBSL 3LINE PIEC BLSL LSLCP 5LELL*,11x/ 6*LSSL 4LINE PIEC SSL USLCP 8UELL*,11x/ 9*UTSL 5LINE PIEC SSL USLCP 8UELL*,11x/ 9*UTSL 6LINE PIEC USLCP TUSL UELL*,21x/ A 1 MAPAX*) 1900 FORMAT(*YUSL YUCL*.66x/ 1 'YLSL YUCL*.66x/ 6 'YUSLCP YSSL*.66x/ A 'ZUSLCP ZUSL*.66x/ B 'ZUSLCP ZUSL*.66x/ C 'ZSSL ZSID*.66x/ C 'ZSSL ZSID*.66x/ END  COMMON /RA/X(10)*Z(10)*TH(10)*H(10)*K(10)*R(10)*B(10)*H(10)*	_		TECE.	•			
0013 FORMAT(*0*) 0014 FORMAT(*0*) 0014 FORMAT(*10.5,F9.5) 1700 FORMAT(*10.5,F9.5) 1100 FORMAT(*10.5,F9.5) 1* 1 6 NONCIRCULAR CROSS SECTION			4F10 F1				
0014 FORMAT(3F10.5,F9.5) 1700 FORMAT(11.78%/ 10 1 6 NONCIRCULAR CROSS SECTION 20 LELL 1ELLI PIEC BLCL SID LSCP 1.21%/ 30 UELL 2ELLI PIEC SID TUCL USCP 1.21%/ 40 LBSL 3LINE PIEC BLSL LSLCP 5LELL.11%/ 60 LSSL 4LINE PIEC LSLCP SSL LELL.10%/ 70 USSL 5LINE PIEC SSL USLCP 8UELL.11%/ 90 UTSL 6LINE PIEC USLCP TUSL UELL.21%/ A 1 MAPAX') 1900 FORMAT(10 USL YUCL.66%/ 1 1 YLSL YUCL.66%/ 6 17 USLCP YSSL.66%/ 7 1 YLSLCP YSSL.66%/ A 2USLCP ZUSL.66%/ 8 2LSLCP ZUSL.66%/ B 2LSLCP ZUSL.66%/ C 2SSL ZSID.66%/ END  SUBROUTINE LIST  LIST QUICK INPUT FILE  COMMON /RA/X(10).Z(10).TH(10).H(10).K(10).R(10).B(10).H(10).			10r 10.3/				
1700 FORMAT(* 1*,78%/ 1* 1 6 NONCIRCULAR CROSS SECTION 2*LELL 1ELLI PIEC BLCL SID LSCP *.21x/ 3*UELL 2ELLI PIEC SID TUCL USCP *.21x/ 4*LBSL 3LINE PIEC BLSL LSLCP 5LELL*,11x/ 6*LSSL 4LINE PIEC LSLCP SSL LELL*,21x/ 7*USSL 5LINE PIEC SSL USLCP 8UELL*,11x/ 9*UTSL 6LINE PIEC USLCP TUSL UELL*,21x/ 1 MAPAX*) 1 MAPAX*) 1 MAPAX*) 1 1 MAPAX*) 1 1 YVLSL YUCL*,66x/ 1 YVLSL YUCL*,66x/ 4 *ZUSLCP ZUSL*,66x/ A *ZUSLCP ZUSL*,66x/ B *ZLSLCP ZLSL*,66x/ C *ZSSL ZSID*,66x/ C *ZSSL ZSID							
1.1 6 NONCIRCULAR CROSS SECTION							
2*LELL 1ELLI PIEC BLCL SID LSCP '.21x/ 3*UELL 2ELLI PIEC SID TUCL USCP '.21x/ 4*LBSL 3LINE PIEC BLSL LSLCP SLELL*,11x/ 6*LSSL 4LINE PIEC LSLCP SSL LELL*,21x/ 7*USSL SLINE PIEC USLCP TUSL UELL*,21x/ 9*UTSL 6LINE PIEC USLCP TUSL UELL*,21x/ 4 1 MAPAX*) 1900 FORMAT(*YUSL YUCL*,66x/ 1 'YLSL YUCL*,66x/ 6 'YUSLCP YSSL*,66x/ 7 'YLSLCP YSSL*,66x/ 8 'ZLSLCP ZUSL*,66x/ 8 'ZLSLCP ZUSL*,66x/ B 'ZLSLCP ZUSL*,66x/ END  SUBROUTINE LIST LIST QUICK INPUT FILE  COMMON /RA/X(10)*Z(10)*TH(10)*H(10)*K(10)*R(10)*B(10)*H(10)*				C0000 05			
3*UELL ZELLI PIEC SID TUCL USCP '.21x/ 4*LBSL 3LINE PIEC BLSL LSLCP  SLELL*,11x/ 6*LSSL 4LINE PIEC LSLCP SSL LELL*,21x/ 7*USSL 5LINE PIEC SSL USLCP  BUELL*,11x/ 9*UTSL 6LINE PIEC USLCP TUSL UELL*,21x/ A 1 MAPAX*)  1900 FORMAT(*YUSL YUCL*,66x/ 1 'YLSL YUCL*,66x/ 6 'YUSLCP YSSL*,66x/ 7 'YLSLCP YSSL*,66x/ A 'ZUSLCP ZUSL*,66x/ B 'ZLSLCP ZUSL*,66x/ C 'ZSSL ZSID*,66x/ C "ZSSL ZSID*,66x/ LST QUICK INPUT FILE  COMMON /RA/X(10)*Z(10)*TH(10)*H(10)*K(10)*R(10)*B(10)*H(10)*							
**LBSL 3LINE PIEC BLSL LSLCP  SLELL**,11X/  6*LSSL 4LINE PIEC LSLCP SSL LELL**,21X/  7*USSL 5LINE PIEC SSL USLCP  BUELL**,11X/  9*UTSL 6LINE PIEC USLCP TUSL UELL**,21X/  A 1 1 MAPAX*)  1900 FORMAT(*YUSL YUCL**,66X/ 1 'YLSL YUCL**,66X/ 6 'YUSLCP YSSL**,66X/ 7 'YLSLCP YSSL**,66X/ A 'ZUSLCP ZUSL**,66X/ B 'ZLSLCP ZUSL**,66X/ C 'ZSSL ZSID**,66X/ C LSSSL ZSID**,66X/  END  **COMMON /RA/X(10)*Z(10)*TH(10)*H(10)*K(10)*R(10)*B(10)*H(10)*							
SLELL*,11x/ 6'LSSL				SID	TUCL	USCP	'+21X/
6*LSSL			PIEC	BLSL	LSLCP		
7*USSL SLINE PIEC SSL USLCP  8UELL*,11X/ 9*UTSL 6LINE PIEC USLCP TUSL UELL*,21X/ A 1 MAPAX*)  1900 FORMAT(*YUSL YUCL*,66X/ 1 *YLSL YUCL*,66X/ 6 *YUSLCP YSSL*,66X/ 7 *YLSLCP YSSL*,66X/ A *ZUSLCP ZUSL*,66X/ C *ZSSL ZSID*,66X/ C *ZSSL ZSID*,66X/ END  SUBROUTINE LIST  LIST QUICK INPUT FILE  COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),H(10),	SLELL',	11X/					
BUELL*,11X/ 9*UTSL	6°LSSL	4LINE	PIEC	LSLCF	9 \$\$L		LELL'.21X/
BUELL*,11X/ 9*UTSL	7*USSL	5L1NE	PIEC	SSL	USLCP		'
9*UTSL 6LINE PIEC USLCP TUSL UELL*,21x/ A 1	8UELL*,	11x/		•			
A			PIEC	USLICE	P TUSL		UELL1.211/
1900 FORMAT(*YUSL YUCL**66X/ 1							
1							
6 'YUSLCP YSSL'.66X/ 7 'YLSLCP YSSL'.66X/ A 'ZUSLCP ZUSL'.66X/ 8 'ZLSLCP ZLSL'.66X/ C 'ZSSL ZSID'.66X) END  SUBROUTINE LIST  LIST QUICK INPUT FILE  COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),H(10),							
7 'YLSLCP YSSL'.66X/ A 'ZUSLCP ZUSL'.66X/ 8 'ZLSLCP ZLSL'.66X/ C 'ZSSL ZSID'.66X) END SUBROUTINE LIST LIST QUICK INPUT FILE  COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),H(10),							
A 'ZUSLCP ZUSL'.66x/ 8 'ZLSLCP ZLSL'.66x/ C 'ZSSL ZSID'.66x) END  SUBROUTINE LIST  LIST QUICK INPUT FILE  COMMON /RA/X(10).Z(10).TH(10).H(10).K(10).R(10).B(10).H(10).							
8 'ZLSLCP ZLSL'.66X/ C 'ZSSL ZSID'.66X) END SUBROUTINE LIST LIST QUICK INPUT FILE COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),H(10),							
C 'ZSSL ZSID'.66X) END SUBROUTINE LIST LIST QUICK INPUT FILE  COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),H(10),							
END  SUBROUTINE LIST  LIST QUICK INPUT FILE  COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),H(10),							
SUBROUTINE LIST  LIST QUICK INPUT FILE  COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),		• 255L	Z21D.	+06X}			
SUBROUTINE LIST  LIST QUICK INPUT FILE  COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),	END						
LIST QUICK INPUT FILE  COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),H(10),	C	********	******	*******	*********	********	
LIST QUICK INPUT FILE COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),	SUBROU	TINE LIST					
COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),	C						
COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),	C LIST QUICK	INPUT FIL	.E				
	C		-				
	COMMON	/RA/X(10)	.Z(10).	TH(10) -F	1(10) .K(10)	-R(10) -B(10	) •H(10) •

```
2 KSI(10).ETA(10).RHO(10).XQ(11).ZQ(11).TQ(11.2).KRV(3).CL(2).
                                                                         140100
    3 KV(2) .V(10,3.3) .SL(3) .RHOSL(10.3) .CRVSL(10.3)
                                                                         140200
    4.VB(10.3.2.8).CRVB (10.8).XMAPAX(10).ZMAPAX(10).XSID(10).ZSID(10).
                                                                         140300
                                                                         140400
    5 BLB(8)
                                                                         140500
     COMMON /IA/ICASE(10).JCASE(30).NCASE(7).NP(3).NS(3).NB(8)
                                                                         140600
     COMMON /RS/KI.MI.KJ.MJ.MIJ.LINE.BLNK.RINF.ERRMAX.OPDN.ELLX.
    1 YUCL.ZERO.FORE.AFTE.TOPS.BOTS.OPUP.OGIV.PIO180.XJ.ZJ.XI.ZI.MI.
                                                                         140700
    2 RI.BI.THI.BJ.A1.A2.LIHT.S.THIJ.S.THIJ.SIJ.AI.HJ.RJ.A4.DX
                                                                         140800
    3.TYPE.ASYM.UPFO.UPAF.UPTO.UPBO.DNFO.DNAF.DNTO.DNBO
                                                                         140900
     COMMON /IS/IR.IW.IF.IP.IPRINT.NSEG.NSEGP1.ISTOP.JSTOP.KSTOP.
                                                                         141000
    1 I.J.L.IP1.LM1.II.NPTS.IZ.ISL.LB.N.NM1.I3
                                                                         141100
     COMMON /LA/SLOPE(10).IDEN(80).CARD(80)
                                                                         141200
     COMMON /LS/TI.T
                                                                         141300
     REAL K.M.KSI.KI.MI.KJ.MJ.MIJ.LINE.KV.KRV
                                                                         141400
     LOGICAL®1 SLOPE.IDEN.CARO.TI.T
                                                                         141500
     IF(IF.EQ.IP)GO TO 9060
                                                                         141600
9000 CONTINUE
                                                                         141700
     REWIND IF
                                                                         141800
     WRITE (1W.9010) IF
                                                                         141900
9010 FORMAT( OLISTING OF QUICK INPUT DECK ON UNIT .12.1:1)
                                                                         142000
     WRITE (IW-2007)
                                                                         142100
9020 READ(IF.90001.END=9040) IDEN
                                                                         142200
     WRITE (IW.2009) IDEN
                                                                         142300
9030 FORMAT( * .80A1)
                                                                         142400
     60 TO 9020
                                                                         142500
9040 WRITE(IW.2007)
                                                                         142600
9050 FORMAT(* *.132(*.*))
                                                                         142700
     WRITE (IW-9055)
                                                                         142800
9055 FORMAT('1')
                                                                         142900
9060 RETURN
                                                                         143000
90001 FORMAT(80A1)
                                                                         143100
2007 FORMAT(* *,82(*-*))
                                                                         143200
2009 FORMAT( 11,80A1, 11)
                                                                         143300
                                                                         143400
143500
     SUBROUTINE SLCE
                                                                         143600
C------
                                                                         143700
 COMPUTÉ SLICE GEOMETRY
                                                                         143800
C-------
                                                                         143900
     COMMON /RA/X(10) . Z(10) . TH(10) . H(10) . K(10) . R(10) . B(10) . M(10) .
                                                                         144000
    1 CURVE(10).DIR(10).X1(10).X2(10).Z1(10).Z2(10).T1(10).T2(10).
                                                                         144100
    2 K5I(10).ETA(10).RHO(10).XQ(11).ZQ(11).TQ(11.Z).KRV(3).CL(2).
                                                                         144200
    3 KV(2) • V(10,3,3) • SL(3) • RHOSL(10,3) • CRVSL(10,3)
                                                                         144300
    4.V8(10.3.2.8),CRV8 (10.8).XMAPAX(10).ZMAPAX(10).XSID(10).ZSID(10).
                                                                         144400
    5 BLB (8)
                                                                         144500
     COMMON /TA/ICASE(10).JCASE(30).NCASE(7).NP(3).NS(3).NB(8)
                                                                         144600
     COMMON /RS/KI.MI.KJ.MJ.MIJ.LINE.BLNK.RINF.ERRMAX.OPDN.ELLX.
                                                                         144700
    1 YUCL.ZERO.FORE.AFTE.TOPS.BOTS.OPUP.OGIV.PIO180.XJ.ZJ.XI.ZI.MI.
                                                                         144800
    Z RI-BI-THI-BJ-A1-A2-A3-BIJ-THJ-S-THIJ-SIJ-AI-HJ-RJ-A4-DX
                                                                         144900
    3.TYPE.ASYM.UPFO.UPAF.UPTO.UPBO.DNFO.DNAF.DNTO.DNBO
                                                                         145000
     COMMON /IS/IR.IW.IF.IP.IPRINT.NSEG.NSEGP].ISTOP.JSTOP.KSTOP.
                                                                         145100
    1 I.J.L.IPI.LMI.II.NPTS.IZ.ISL.LB.N.NMI.I3
                                                                         145200
     COMMON /LA/SLOPE(10).IDEN(80).CARD(80)
                                                                         145300
     COMMON /LS/TI.T
                                                                         145400
     REAL K.M.KSI.KI.MI.KJ.MJ.MIJ.LINE.KV.KRV
                                                                         145500
     LOGICAL®1 SLOPE.IDEN.CARD.TI.T
                                                                         145600
```

10000 CONTINUE	145700
IF(TYPE.NE.ASYM)GO TO 10006	145800
C SET UP XQ AND ZQ FOR ASYM CASE	145900
NSEGP1=1	146000
XQ(1)=VB(NB(1)+1+1+1+1)	146100
A1=0.	146200
DO 10005 L=1.3	146300
11=N8 (L) +1	146400
DO 10005 I=1.I1	146500
A1=AMAX1 (A1.VB(I.2.1.L))	146600
10005 CONTINUE	146700
ZQ(1)=A1	146800
10006 CONTINUE	146900
C DETERMINE IF SLICES ARE PRESENT	147000
ISL=0	147100
DO 10010 I=1.10	147200
DO 10010 J=1.3	147300
DO 10010 L=1.3	147400
IF(V(I.J.L).GT.RINE)GO TO 10010	147500
ISL=1	147600
GO TO 10015	147700
10010 CONTINUE	147800
IF(TYPE.EQ.ASYM.AND.ISL.EQ.O)GO TO 10015	147900
IF (ISL.EQ.O) RETURN  C. DETERMINE NUMBER OF CERMINE IN EACH SLICE NEW YORK AND ALL AN	148000
C DETERMINE NUMBER OF SEGMENTS IN EACH SLICE NS(L)=NP(L)=1 10015 DO 10020 L=1.3	148100
NP(L)=0	148200
DO 10020 I=1-10	148300 148400
DO 10020 J=1.3	148500
	148600
IF(V(I.J.L).LT.RINF)NP(L)=1 10020 CONTINUE	148700
00 10030 L=1.3	148800
NS(L)=NP(L)-1	148900
10030 CONTINUE	149000
C CHECK INPUT FOR SUFFICIENCY	149100
DO 10060 L=1.3	149200
11=NP(L)	149300
IF(I)-E9.0)80 TO 10060	149400
DO 10055 1-1-11	149500
12-0	149600
DO 10040 J=1,3	149700
IF(V(I.J.L).LT.RINF)12=12+1	149800
10040 CONTINUE	149900
IF(12.GE.2)60 TO 10055	150000
ISTOP=1	150100
WRITE(IW-10050)L-I	150200
10050 FORMAT( COND CODE 0050: SLICE .12 POINT .13 DATA INSUFFICIE	
INT*)	150400
10055 CONTINUE	150500
10060 CONTINUE	150600
IF(ISTOP.EQ.0)GO TO 10070	150700
ūritē (iw∙2035)	150800
2035 FORMAT(* EXECUTION HALTED*)	150900
STOP 0050	151000
C COMPUTE REMAINING SLICE GEOMETRY	151100
10070 DO 10160 L=1.3	151200

### DC-T R-77-89

Table 8. Continued

```
Il=NP(L)
                                                                             151300
      IF(I1.EQ.0)GO TO 10160
                                                                             151400
      DO 10150 I=2.I1
                                                                             151500
      IM1=I-1
                                                                             151600
      DO 10080 Jel.3
                                                                             151700
      IF(V(I,J,L).GT.RINF)GO TO(10100,10130,10140).J
                                                                             151800
10080 CONTINUE
                                                                             151900
      Al=(V(I.2.L)=V(IM1.2.L))/(V(I.1.L)=V(IM1.1.L))
                                                                             152000
      Al=ATAN(A1)/PIO180
                                                                             152100
      IF (ABS(A1-V(1.3.L)).LE.ERRMAX)GO TO 10150
                                                                             152200
      ISTOP=1
                                                                             152300
      WRITE(IW.10090)L.I.ERRMAX
                                                                             152400
10090 FORMAT( COND CODE 0090: SLICE .12. . POINT .13. . REDUNDANT INPUT
                                                                             152500
     1 DATA NOT SELF CONSISTENT TO WITHIN .1PE15.7)
                                                                             152600
      GO TO 10150
                                                                             152700
10100 IF(V(1.3.L).EQ.0.)60 TO 10110
                                                                             152800
      V(I-1-L)=(V(I-2-L)-V(IM1-2-L))/TAN(PIO180+V(I-3-L))+V(IM1-1-L)
                                                                             152900
      GO TO 10150
                                                                             153000
10110 WRITE(IW-10120)L-I
                                                                             153100
10120 FORMAT ( COND CODE 0120: SLICE , 12. . . POINT , 13. . . X MUST BE INPUT
                                                                             153200
     1 WHEN SLOPE IS ZERO*)
                                                                             153300
      WRITE (IW, 2035)
                                                                             153400
      STOP 0120
                                                                             153500
10130 V(I+2+L)=V(IM1+2+L)+(V(I+1+L)-V(IM1+1+L))+TAN(PI0180+V(I+3+L))
                                                                             153600
      GO TO 10150
                                                                             153700
10140 V(I,3.L)=ATAN((V(I,2.L)-V(IM1,2.L))/(V(I,1.L)-V(IM1,1.L)))/PIO180
                                                                             153800
10150 CONTINUE
                                                                             153900
10160 CONTINUE
                                                                             154000
      IF(ISTOP.EQ.0)GO TO 10170
                                                                             154100
      WRITE (IW.2035)
                                                                             154200
      STOP 0090
                                                                             154300
C EXTRAPOLATE BEGINNING OF FIRST SEGMENT TO X=0
                                                                             154400
10170 DO 10180 L=1,3
                                                                             154500
      II=NP(L)
                                                                             154600
                                                                             154700
      IF(I1.EQ.0)GO TO 10180
      IF(V(1.1.L).LE.O.) GO TO 10180
                                                                             154800
      V(1,2,L)=V(1,2,L)=V(1,1,L)*TAN(PIO180*V(2,3,L))
                                                                             154900
      V(1.1.L)=0.
                                                                             155000
10180 CONTINUE
                                                                             155100
C EXTRAPOLATE END OF LAST SEGMENT TO XQ(NSEGP1)
                                                                             155200
      DO 10190 L=1,3
                                                                             155300
      11=NP(L)
                                                                             155400
      IF(I1.EQ.0)GO TO 10190
                                                                             155500
      IF(V(II+1+L).GE.XQ(NSEGP1))GO TO 10190
                                                                             155600
      V(11-2-L)=V(11-2-L)+(XQ(NSEGP1)+V(11-1-L))+TAN(PIO180+V(11-3-L))
                                                                             155700
      V(11.1.L) = XQ(NSEGP1)
                                                                             155800
10190 CONTINUE
                                                                             155900
C PLACE NON EXISTENT SLICES OUTSIDE OF VEHICLE
                                                                             156000
      A1=0.
                                                                             156100
      DO 10200 I=1,NSEGP1
                                                                             156200
      Al=AMAX1(Al.ABS(ZQ(I)))
                                                                             156300
10200 CONTINUE
                                                                             156400
      A1=2.-A1
                                                                             156500
      DO 10210 L=1.3
                                                                             156600
      IF (NP(L).NE.0) 60 TO 10210
                                                                             156700
      NP (L) =2
                                                                             156800
```

\_K.

```
....
                .:/:
     NS(L)=1
                                                                      156900
     V(1-1-L)=0."
                                                                      157000
     V(1.2.L)=A1
                                                                      157100
     V(2.1.L)=XQ(NSEGP1)
                                                                      157200
     V(2.2.L)=A1
                                                                      157300
     V(1.3.L)=0.
                                                                      157400
     V(2+3+L)=0.
                                                                      157500
10210 CONTINUE
                                                                      157600
C BE SURE V(1.3.L) IS DEFINED
                                                                      157700
     DO 10220 L=1.3
                                                                      157800
     IF (V(1.3.L).GT.RINF) V(1.3.L)=V(2.3.L)
                                                                      157900
10220 CONTINUE
                                                                      158000
C PRINT RESULTS OF SLICE GEOMETRY CALCULATIONS
                                                                      158100
     WRITE(IW.10230)
                                                                      158200
10230 FORMAT(*OSLICE GEOMETRY:*/'0*,21x.*UPPER*,39x.*LOWER*,39x.*SIDE *
                                                                      158300
    1/ 9x • 'X • • 13x • 'Z • • 13x • 'T • • 15x • 'X • • 13x • 'Z • • 13x • *T • • 15x • *X • • 13x •
                                                                      158400
    2'Y'+13X+'T'/' '+132('-'))
                                                                      158500
     DO 10250 I=1.10
                                                                      158600
     WRITE(IW.10240) I.((V(I.J.L).J=1.3).L=1.3)
                                                                      158700
10240 FORMAT( 1,12,2(1P3E14.6,2x),1P3E14.6)
                                                                      158800
10250 CONTINUE
                                                                      158900
C CHECK VALIDITY OF COORDINATES
                                                                      159000
     DO 10280 L=1.3
                                                                      159100
     Il=NP(L)
                                                                      159200
     DO 10270 I=1.11
                                                                      159300
     IF(V(I.2.L).GE.0.)GO TO 10270
                                                                      159400
     ISTOP=1
                                                                      159500
     WRITE(IW+10260)L+I
                                                                      159600
10260 FORMAT( COND CODE 0260: SLICE , 12, , POINT , 13, , Z VALUE < 0. SL
                                                                     159700
    lices MAY NOT CROSS VEHICLE CL. ")
                                                                      159800
10270 CONTINUE
                                                                      159900
10280 CONTINUE
                                                                      160000
     IF (ISTOP.EQ.0)GO TO 10290
                                                                      160100
     WRITE(IW.2035)
                                                                      160200
     STOP 0260
                                                                      160300
C RESIGN LSL FOR QUICK INPUT
                                                                      160400
10290 I1=NP(2)
                                                                      160500
     IF (TYPE-EQ-ASYM) RETURN
                                                                      160600
     DO 10300 I=1.I1
                                                                      160700
     DO 10300 J=2.3
                                                                      160800
     V(I+J+Z)==V(I+J+Z)
                                                                     160900
10300 CONTINUE
                                                                      161000
     RETURN
                                                                     161100
                                                                     161200
161300
     SUBROUTINE RESL
                                                                     161400
161500
C INSERT ROUNDS OR FILLETS BETWEEN SEQUENTIAL SLICING PLANES
                                                                     161600
161700
     COMMON /RA/X(10) .Z(10) .TH(10) .H(10) .K(10) .R(10) .B(10) .M(10) .
                                                                     161800
    1 CURVE(10).DIR(10).X1(10).X2(10).Z1(10).Z2(10).T1(10).T2(10).
                                                                     161900
    2 KSI(10) .ETA(10) .RHO(10) .XQ(11) .ZQ(11) .TQ(11.2) .KRY(3) .CL(2) .
                                                                     162000
    3 KV(2) .V(10.3,3) .SL(3) .RHOSL(10.3) .CRVSL(10.3)
                                                                     162100
    4.V8(10.3.2.8).CRV8 (10.8).XMAPAX(10).ZMAPAX(10).XSID(10).ZSID(10).
                                                                     162200
    5 BLB(8)
                                                                     162300
```

COMMON /IA/ICASE(10).JCASE(30).NCASE(7).NP(3).NS(3).NB(8)

162400

168000

```
COMMON /R5/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
                                                                              162500
     1 YUCL, ZERO, FORE, AFTE, TOPS, BOTS, OPUP, OGIV, PIO180, XJ, ZJ, XI, ZI, HI,
                                                                              162600
     Z RI-BI-THI-BJ-Al-AZ-AZ-BIJ-THJ-S-THIJ-SIJ-AI-HJ-RJ-A4-DX
                                                                              162700
      3. TYPE, ASYM, UPFO, UPAF, UPTO, UPBO, DNFO, DNAF, DNTO, DNBO
                                                                              162800
      COMMON /IS/IR.IV.IF.IP.IPRINT.NSEG.NSEGP1.ISTOP.JSTOP.KSTOP.
                                                                              162900
      1 1.J.L.IP1.LM1.II.NPTS.IZ.ISL.LB.N.NM1.I3
                                                                              163000
      COMMON /LA/SLOPE(10) . IDEN(80) . CARD(80)
                                                                              163100
      COMMON /LS/TI.T
                                                                              163200
      REAL KOMOKSIOKIOMIOKJOMJOMIJOLINEOKYOKRY
                                                                              163300
      LOGICAL+1 SLOPE, IDEN, CARD, TI, T
                                                                              163400
11000 CONTINUE
                                                                              163500
      1SKIP=0
                                                                              163600
      IF (ISL.EQ.0) RETURN
                                                                              163700
C DETERMINE IF ROUNDS OR FILLETS ARE PRESENT
                                                                              163800
      IRFSL =0
                                                                              163900
      DO 11005 L=1.3
                                                                              164000
      DO 11005 I=1.10
                                                                              164100
      IF(RHOSL(I.L).GT.RINF)GO TO 11005
                                                                              164200
      IRFSL=1
                                                                              164300
      60 TO 11006
                                                                              164400
11005 CONTINUE
                                                                              164500
      RETURN
                                                                              164600
11006 WRITE(IW-11007)
                                                                              164700
11007 FORMAT ( OSLICE ROUND/FILLET CENTERS AND INTERSECTIONS:
                                                                              164800
            /101, 9x. 1811-11X. 18J1-11X. 1A31-11X. 1441-11X. 1X11-11X, 1Z1.
                                                                              164900
                 11x+'T1'+11x+'X2'+11x+'Z2'+11x+'T2'/'
                                                                              165000
     2 132(1-1))
                                                                              165100
C COMPUTE ROUND/FILLET CENTERS AND INTERSECTIONS
                                                                              165200
      00 11130 L=1.3
                                                                              165300
      DO 11030 I=2.9
                                                                              165400
      IP1=I+1
                                                                              165500
      IF (RHOSL(I.L).GT.RINF)GO TO 11030
                                                                              165600
      ISKIP=1
                                                                              165700
      IF(ABS(V(I,3,L)-V(IP1,3,L)).GE.ERRMAX)GO TO 11020
                                                                              165800
      WRITE(IW-11010)1-IP1
                                                                              165900
11010 FORMAT ( COND CODE 1010: FILLETS/ROUNDS MAY NOT BE INSERTED BETWEE
                                                                             166000
     IN LINE SEGMENTS WITH SAME SLOPE, SEGMENTS . 213)
                                                                              166100
      WRITE (IW.2035)
                                                                              166200
 2035 FORMAT ( * EXECUTION HALTED )
                                                                              166300
      STOP 1010
                                                                              166400
11020 S=1.
                                                                              166500
      IF (V(IP1.3.L).GT.V(I.3.L))S=-1.
                                                                              166600
      MI=TAN(PIO180-V(I,3,L))
                                                                              166700
      MJ=TAN(PIO180+V(1P1.3.L))
                                                                              166800
      A1=V(I,2,L)-MI+V(I,1,L)
                                                                              166900
      A2=V(IP1,2,L)-MJ=V(IP1,1,L)
                                                                              167000
      BI=A1-S-RHOSL(I.L)/COS(PIO180+V(I.3.L))
                                                                              167100
      BJ=A2-S=RHOSL(I+L)/COS(PIO180+V(IP1+3+L))
                                                                             167200
      (IM-LM) \ (LB+IM-IB+LM) = EA
                                                                             167300
      (IM-LM) / (IB-LB) -=+4A
                                                                             167400
      X1(I)=A4-S-RHOSL(I+L)-SIN(PIO180+V(I+3+L))
                                                                             167500
      X2(I)=A4-S-RHOSL(I+L)+SIN(PIO180+V(IP1,3+L))
                                                                             167600
      Z1(I)=A3.S*RHOSL(I,L)+CUS(PIO180*V(I,3,L))
                                                                             167700
      ZZ(I)=A3+S+RHOSL(I+L)+COS(PIO180+V(IP1+3+L))
                                                                             167800
      T1(I)=V(I-3-L)
                                                                             167900
```

T2(I)=V(IP1,3,L)

write(iw.11025)L.i.Bi.Bj.BJ.A3.A4.x1(I).71(I).71(I).X2(I).72(I).472(I)	168100
11025 FORMAT(1X.211.1P10E13.5)	168200
11030 CONTINUE	168300
C INSERT ROUNDS/FILLETS INTO QUICK ARRAYS	168400
DO 11035 I-1.9	168500
	168600
IP1=1+1	
RHOSL(I+L)=RHOSL(IP1+L)	168700
X1(I)=X1(IP1)	168800
Z1(I)=Z1(IP1)	168900
T1(I)=T1(IP1)	169000
x2(1)=x2((P1)	169100
72(1)=22(IP1)	169200
	169300
T2(1)=T2(IP1)	
11035 CONTINUE	169400
RHOSL(10,L)=1.E70	169500
J=0	169600
DO 11110 I=1.10	169700
Jm.je1	169800
IF(RHOSL(I,L).GT.RINF)GO TO 11110	169900
	170000
J#J+1	170100
NS (L) =NS (L) +1	
NP(L)=NP(L)+1	170200
IF(NP(L).LE.10)GO TO 11050	170300
WRITE(IW-11040)I	170400
11040 FORMAT(* COND CODE 1040: INSERTION OF ROUND/FILLET*, 13, * EXCEEDS D	170500
11MENSIONS (10)*)	170600
WRITE (IW-2035)	170700
STOP 1040	170800
	170900
11050 N=11	171000
11060 N=N-1	
NM1=N-1	171100
IF(N.LE.J)GO TO 11080	171200
DO 11070 I2=1.3	171300
11070 V(N.I2.L)=V(NM1.I2.L)	171400
CRVSL (N+L)=CRVSL (NM1+L)	171500
60 TO 11060	171600
11080 CONTINUE	171700
	171800
V(J-1-L)=X1(I)	
V(J-2-L)=Z1(I)	171900
V(J,3,L)=T1(I)	172000
CRVSL (J+1,L)=CRVSL (J,L)	172100
V(J-1-1-L)=X2(I)	172200
V(J-1-2-L)=Z2(I)	172300
V(J+1,3,L) =T2(1)	172400
CRVSL(J+L)=ELLX	172500
	172600
11110 CONTINUE	
WRITE(IW+11120)	172700
11120 FORMAT(* *)	172800
11130 CONTINUE	172900
C PRINT FINAL QUICK ARRAYS	173000
WRITE(IW-10230)	173100
10230 FORMAT( OF INAL QUICK ARRAYS FOR ROUNDED/FILLETTED SLICES! .	173200
1 /*0*,21x,*UPPER*,39X,*LOWER*,39X,*SIDE *	173300
	173400
1/ 9x.·x·.13x.·z· .13x.·r·.15x.·x·.13x.·z·.13x.·r·.15x.·x·.13x.	173500
2'4',13X,17',' ',132('-'))	
DO 10250 I=1.10	173600

Table 8. Continued

```
WRITE(IW-10240)I+((V(I-J-L)-J-1,3)-L-1,3)
                                                                         173700
10240 FORMAT( 1,12,2(1P3E14.6,2X),1P3E14.6)
                                                                         173800
10250 CONTINUE
                                                                         173900
      RETURN
                                                                         174000
      END
                                                                         174100
174200
      SUBROUTINE COPY
                                                                         174300
C------
                                                                         174400
C COPY DATA INTO VB ARRAY FOR QK3D
                                                                         174500
174600
      COMMON /RA/X(10)+Z(10)+TH(10)+H(10)+K(10)+R(10)+B(10)+H(10)+
                                                                         174700
     1 CURVE(10) .DIR(10) .X1(10) .X2(10) .Z1(10) .Z2(10) .T1(10) .T2(10) .
                                                                         174800
     2 KSI(10) .ETA(10) .RHO(10) .XQ(11) .ZQ(11) .TQ(11.2) .KRV(3) .CL(2) .
                                                                         174900
     3 KV(2) +V(10+3+3) +SL(3) +R-OSL(10+3) +CRYSL(10+3)
                                                                         175000
     4. VB(10.3.2.8).CRVB (10.8), XMAPAX(10), ZHAPAX(10), XSID(10), ZSID(10).
                                                                         175100
     5 BLB(8)
                                                                         175200
      COMMON /IA/ICASE(10).JCASE(30).NCASE(7).NP(3).NS(3).NB(8)
                                                                         175300
      COMMON /RS/KI.MI.KJ.MJ.MIJ.LINE.BLNK.RINF.ERRMAX.OPDN.ELLX.
                                                                         175400
     1 YUCL, ZERO, FORE, AFTE, TOPS, BOTS, OPUP, OGIV, PIO180, XJ, ZJ, XI, ZI, HI,
                                                                         175500
     Z RI-BI-THI-BJ-A1-A2-A3-BIJ-THJ-S-THIJ-SIJ-A1-HJ-RJ-A4-DX
                                                                         175600
     3. TYPE. ASYM. UPFO. UPAF. UPTO. UPBO. DNFO. DNAF. DNTO. DNBO
                                                                         175700
      COMMON /IS/IR.IW.IF.IP.IPRINT.NSEG.NSEGP1.ISTOP.JSTOP.KSTOP.
                                                                         175800
     1 I.J.L.IP1,LM1.II,NPTS.I2.ISL.LB.N.NM1.I3
                                                                         175900
      COMMON /LA/SLOPE(10) . IDEN(80) . CARD(80)
                                                                         176000
      COMMON /LS/TI.T
                                                                         176100
      REAL K.M.KSI.KI.HI.KJ.MJ.MIJ.LINE.KV.KRV
                                                                         176200
      LOGICAL®1 SLOPE.IDEN.CARD.TI.T
                                                                         176300
      DIMENSION E(10.7)
                                                                         176400
      EQUIVALENCE (E(1,1).x (1))
                                                                         176500
                 .(E(1.2).Z (1))
                                                                         176600
                +(E(1.3).TH(1))
                                                                         176700
     3
                 +(E(1.4)+H (1))
                                                                         176800
                 .(E(1.5).K (1))
                                                                         176900
                 .(E(1.6).R (1))
                                                                         177000
                .(E(1,7),B (1))
                                                                         177100
      DATA RINF1/1.E70/
                                                                         177200
      LOGICAL®1 F/'F'/
                                                                         177300
12000 CONTINUE
                                                                         177400
                                                                         177500
      60 TO(12010,12010,12010,12050,12050,12050,12080,12080
                                                                         177600
     1) +L8
                                                                         177700
C L8=1 ZUCL, L8=2 ZLCL, L8=3 YSID
                                                                         177800
12010 NB(LB)=NSEG
                                                                         177900
      DO 12020 I=1.10
                                                                         178000
      VB(I+1+1+LB)=XQ(I)
                                                                         178100
      VB(1.2.1.LB)=ZQ(1)
                                                                         178200
      VB(I,3,1,LB)=TQ(I,1)
                                                                         178300
      VB(1.3.2.LB)=TQ(1.2)
                                                                         178400
      CRVB (I,LB)=CURVE(I)
                                                                         178500
12020 CONTINUE
                                                                         178600
      DO 12040 I=1,10
                                                                         178700
      XQ(I)=RINF1
                                                                         178800
      ZQ(I) #RINF1
                                                                         178900
      TQ([.1)=RINF1
                                                                         179000
      TQ(I.2)=RINF1
                                                                         179100
      00 12030 J=1.7
                                                                         179200
```

E(I,J)=RINF1	179300
12030 CONTINUE	179400
SLOPE(I)=F_	179500
RHO(I)=RINF1	179600
X1(1)=RINF1	179700
X2(I)=RINF1	179800
Z1(I)=RINF1	179900
Z2(1)=RINF1	180000
T1(1)=RINF1	180100
T2(I)=RINF1	180200
KSI(1)=RINF) '	180300 180400
ETA(I)=RINF1 CURVE(I)=BLNK	180500
DIR(I)=DPDN	180600
12040 CONTINUE	180700
RETURN .	180800
C LB=4 ZUSL+ LB=5 ZLSL+ LB=5 YSSL	180900
12050 11=LB-3	181000
NB(LB)=NS(I1)	181100
DO 12070 1=1+10	181200
DO 12060 J=1.3	181300
VB(I.J.1.LB)=V(I.J.II)	181400
CRVB(1.LB)=CRVSL(1.11)	181500
12060 CONTINUE	181600
IF(I.LT.10)VB(1.3.2.LB)=V(I.1.3.I1)	181700
12070 CONTINUE	181800
RETURN	181900
12080 DO 12090 I=1.10	182000
VB(I,1,1,7)=XSID(I)	182100
VB(I+2+1+7)=ZSID(I)	182200
VB(I:1:1:8) =XMAPAX(I)	182300
VB(I+2+1+8)=ZMAPAX(I)	182400
12090 CONTINUE	182500
RETURN	182600
_ END	182700
C	
SUBROUTINE BLCK	182900
	183000
C CHECK ALL BODY LINE COORDINATES FOR VALIDITY	183100
C COMPUTE DEFAULT VALUES FOR ZSID AND ZMAPAX	183200 183300
COMMON /RA/X(10) • Z(10) • TH(10) • H(10) • K(10) • R(10) • B(10) • H(10) •	183400
1 CURVE(10).DIR(10).X1(10).X2(10).Z1(10).Z2(10).T1(10).T2(10).	
2 KSI(10) •ETA(10) •RHO(10) •XQ(11) •ZQ(11) •TQ(11•2) •KRV(3) •CL(2)	
3 KV(2) • V(10•3•3) • SL(3) • RHOSL(10•3) • CRYSL(10•3)	183700
4.VB(10.3.2.8).CRVB (10.8).XMAPAX(10).ZMAPAX(10).XSID(10).ZSI	
5 BLB(8)	183900
COMMON /IA/ICASE(10).JCASE(30).NCASE(7).NP(3).NS(3).NB(8)	184000
COMMON /RS/KI.MI.KJ.MJ.MIJ.LINE.BLNK.RINF.ERRMAX.OPDN.ELLX.	184100
1 YUCL.ZERO.FORE.AFTE.TUPS.BOTS.OPUP.OGIV.PI0180.XJ.ZJ.XI.ZI.I	
2 KI.BI.THI.BJ.AI.AZ.A3.BIJ.THJ.S.THIJ.SIJ.AI.HJ.RJ.A4.DX	184300
3.TYPE.ASYM.UPFO.UPAF.UPTO.UPBO.DNFO.DNAF.DNTO.DNBO	184400
COMMON /IS/IR.IW.IF.IP.IPRINT.NSEG.NSEGPI.ISTOP.JSTOP.KSTOP.	184500
1 I.J.L. IP1.LM1.II.NPTS.IZ.ISL.LB.N.NM].I3	184600
COMMON /LA/SLOPE(10) . IDEN(80) . CARD(80)	184700
COMMON /LS/TI.T	184800

190400

```
Table 8. Continued
       REAL K.M.KSI.KI.MI.KJ.MJ.MIJ.LINE.KV.KRV
                                                                             184900
       LOGICAL*1 SLOPE.IDEN.CARD.TI.T
                                                                             185000
13000 CONTINUE
                                                                             185100
C ZSID AND ZMAPAX DEFAULT
                                                                             185200
      DO 13060 Le7.8
                                                                             185300
       NB(L)=0
                                                                             185400
      DO 13010 I=1.10
                                                                             185500
       IF (VB(I+1+1+L).LT.RINF.AND.VB(I+2+1+L).LT.RINF)NB(L)=I
                                                                             185600
13010 CONTINUE
                                                                             185700
      NB(L)=NB(L)+1
                                                                             185800
      IF (NB(L).GT.0)GO TO 13060
                                                                             185900
      IF(NB(L).EQ.-1)GO TO 13030
                                                                             186000
      WRITE(IW-13020)L
                                                                             186100
13020 FORMAT ( COND CODE 3020: BODY LINE . 12. REQUIRES X AND Z VALUES F
                                                                             186200
     10R TWO OR MORE POINTS!)
                                                                             186300
      WRITE(IW-2035)
                                                                             186400
 2035 FORMAT (* EXECUTION HALTED*)
                                                                             186500
      STOP 3020
                                                                             186600
13030 IF (L.NE.7) GO TO 13040
                                                                             186700
      NB (7) =1
                                                                             186800
      VB(1+1+1+7)=0.
                                                                             186900
      VB(1-2-1-7)=0.
                                                                             187000
      VB(2-1-1-7)=XQ(NSE6P1)
                                                                             187100
      VB(2.2.1.7)=0.
                                                                             187200
      GO TO 13060
                                                                             187300
13040 I1=NB(7)+1
                                                                             187400
      NB(8)=NB(7)
                                                                             187500
      DO 13050 I=1.I1
                                                                             187600
      DO 13050 Jel.2
                                                                             187700
      VB(1,J,1,8)=VB(1,J,1,7)
                                                                             187800
13050 CONTINUE
                                                                             187900
13060 CONTINUE
                                                                             188000
C VALIDITY CHECK
                                                                             188100
      A1=VB(NB(1)+1.1.1.1)
                                                                             188200
      DO 13080 L=2.8
                                                                             188300
      I1=NB(L)+1
                                                                             188400
      IF (ABS(VB(I1.1.1.L)-A)).LE.ERRMAX)GO TO 13080
                                                                             188500
      IF(VB(I1-1-1-L).GE.A1.AND.L.GE.4)GO TO 13080
                                                                             188600
      WRITE(IW+13070)ERRMAX+L
                                                                             188700
13070 FORMAT( COND CODE 3070: ALL BODY LINES MUST BE THE SAME LENGTH TO
                                                                            188800
     1 WITHIN ERRMAX=", 1PE14.7. BL .. [2]
                                                                             188900
      WRITE (IW.2035)
                                                                             189000
      STOP 3070
                                                                             189100
13080 CONTINUE
                                                                             189200
      DO 13100 L=1.8
                                                                             189300
      IF (ABS(VB(1+1+1+L)).LE.ERRMAX)GO TO 13100
                                                                             189400
      WRITE(IW-13090) ERRMAX-L-VB(1,1-1-L)
                                                                             189500
13090 FORMAT( COND CODE 3090; FIRST X OF EACH BODY LINE MUST BE ZERO (T
                                                                           189600
     10 WITHIN ERRMAX=++1PE14.7,+) BL++12. * X=+,1PE14.7)
                                                                            189700
      WRITE(IW-2035)
                                                                             189800
      STOP 3090
                                                                             189900
13100 CONTINUE
                                                                             190000
C RESIGN ZLCL AND ZLSL
                                                                             190100
      DO 13110 L=2.5.3
                                                                            190200
      11=NB(L)+1
                                                                            190300
```

DO 13110 I=1.I1

	00 13110 4-0 3	
	00 13110 J=2.3	190500
	DO 13110 N=1.2	190600
	IF(VB(I+J+N+L).GT.RINF)GO TO 13110	190700
	VB(I+J+N+L) =-VB(I+J+N+L)	190800
13110	CONTINUE	190900
C SET	UP ANGLES FOR ZHAPAX AND ZSID	191000
	DO 14026 L=7.8	191100
	11=NB(L)	191200
	DO 14025 I=1.I1	191300
	Al=ATAN((VB(I+1+2+1+L)-VB(I+2+1+L))/(VB(I+1+1+1+L)-VB(I+1+1+L)))	191400
	Al=Al/PIO180	191500
	VB(1+3+1+L)=A1	191600
14036	VB(1,3,2,L)=A1	191700
	CONTINUE	191800
14050	CONTINUE	191900
	RETURN	192000
_	END	192100
Coses		192200
	SUBROUTINE QK3D	192300
C	+44000	192400
	UP QUICK MODEL FOR 3D CASE	192500
C	977-9-V-00-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-	192600
	COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),	192700
	1 CURVE(10).DIR(10).X1(10).X2(10).Z1(10).Z2(10).T1(10).T2(10).	192800
	2 KSI(10).ETA(10).RHO(10).XQ(11).ZQ(11).TQ(11.2).KRV(3).CL(2).	192900
	3 KV(2) •V(10•3•3) •SL(3) •R→OSL(10•3) •CRVSL(10•3)	193000
	4.VB(10.3.2.8),CRVB(10.8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),	193100
	5 BLB(8)	193200
	COMMON /IA/ICASE(10).JCASE(30).NCASE(7).NP(3).NS(3).NB(8)	193300
	COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX.	193400
	1 YUCL.ZERO,FORE,AFTE,TOPS,80TS,0PUP,06IV,PI0180,XJ,ZJ,XI,ZI,MI,	193500
	2 RI.BI.THI.BJ.AI.AZ.A3.BIJ.THJ.S.THIJ.SIJ.AI.HJ.RJ.A4.DX	193600
	3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO	193700
	COMMON /IS/IR.IW.IF.IP.IPRINT.NSEG.NSEGP1.ISTOP.JSTOP.KSTOP.	193800
	1 I.J.L. IP1.LM1.I1.NPTS.I2.ISL.LB.N.NM1.I3	193900
	COMMON /LA/SLOPE(10).IDEN(80).CARD(80)	194000
	COMMON /LS/TI+T	194100
	REAL K-M-KSI-KI-MI-KJ-MJ-MIJ-LINE-KV-KRV	194200
	LOGICAL®1 SLOPE.IDEN.CARD.TI.T	194300
14000	CONTINUE	194400
	IF(IF.NE.IP)REWIND IF	194500
Ç CS	MODELS	194600
	WRITE(IF.90001)IDEN	194700
	WRITE(IF,21700)	194800
	11=1	194900
	WRITE(IF.90004)I].11.ZERO.VB(NB(1)+1.1.1)	195000
C BL	MODELS	195100
	WRITE (IF, 90005) YUCL	195200
	WRITE(IF.90006)[1.KRV(1).KV(1)	195300
	WRITE(IF,90007)ZERO,ZERU,VB(NB(1)+1,1,1,1),ZERO,ZERO,ZERO	195400
	WRITE(1F.9009)	195500
	DO 14020 L=1,8	195600
	11=NB(L)	195700
	WRITE (IF+90005) BLB (L)	195800
	DO 1401G I=1.I1	195900
	IP1=I+1	196000

Table 8. Continued

```
12-1
                                                                                196100
       IF (CRVB(I+L).EQ.ELLX) 12=2
                                                                                196200
       WRITE(IF.90006) I,CRVB(I,L),KV(IZ)
                                                                                196300
      WRITE(IF.90007) VB(I.1.1.L).VB(I.2.1.L).VB(IP1.1.1.L).VB(IP1.2.1.L)
                                                                                196400
     1.VB(1.3.1.L).VB(1.3.2.L)
                                                                                196500
14010 CONTINUE
                                                                                196600
      WRITE(IF,90009)
                                                                                196700
14020 CONTINUE
                                                                                196800
      WRITE (IF.21900)
                                                                                196900
      WRITE (IF.90011)
                                                                                197000
       IF (IPRINT.EQ.0) GO TO 14030
                                                                               197100
C SET UP CARDS TO MAKE QUICK EXERCISE MODEL
                                                                                197200
      DX=.001
                                                                                197300
      A1=DX
                                                                                197400
      A2=VB(NB(1)+1+1+1+1)-DX
                                                                                197500
      A3=(A2-A1) -.01
                                                                                197600
      X0*.5+SA=SA
                                                                                197700
      11-1
                                                                                197800
      12=2
                                                                                197900
      WRITE(IF,90012) I1, I2, A1, A2, A3
                                                                                198000
14030 WRITE(IF, 90013)
                                                                                198100
      RETURN
                                                                                198200
90001 FORMAT(80A1)
                                                                                198300
90004 FORMAT (212,6x,2F10.5)
                                                                                198460
90005 FORMAT (A4)
                                                                                198500
90006 FORMAT(12.1x.A4. PIEC ".A4)
                                                                                198600
90007 FORMAT (3F10.5,2(F9.5,'A').F10.5)
                                                                                198700
90009 FORMAT (*-1*)
                                                                                198800
90011 FORMAT(* *)
                                                                                198900
90012 FORMAT(12,13,5x,6f10.5)
                                                                                199000
90013 FORMAT(* 0*)
                                                                                199100
90014 FORMAT(3F10.5.F9.5)
                                                                                199200
21700 FORMAT(* 1*.78X/
                                                                                199300
     1 1 6
                  NONCIRCULAR CROSS SECTION
                                                                       1×15+1
                                                                               199400
     2'LELL
                 IELLI PIEC
                                 BLCL
                                             SID
                                                        LSCP
                                                                       ..21x/
                                                                               199500
     3ºUELL
                 ZELLI PIEC
                                  SID
                                            TUCL
                                                        USCP
                                                                       1.21X/
                                                                               199600
     4ºLBSL
                 3LINE PIEC
                                 BLSL
                                             LSLCP
                                                                                199700
     5LELL +11X/
                                                                                199800
     6'LSSL
                 4LINE PIEC
                                  LSLCP
                                             SSL
                                                                  LELL'+21X/
                                                                               199900
     7ºUSSL
                 5LINE
                        PIEC
                                  SSL
                                             USLCP
                                                                                200000
     8UELL . 11X/
                                                                                200100
     9ºUTSL
                        PIEC
                 6LINE
                                  USLCP
                                            TUSL
                                                                  UELL .. 21X/
                                                                               200200
                         MAP 1)
                                                                                200300
21900 FORMAT ( YUSL
                         YUCL . 66X/
                                                                                200400
                         YUCL 1 + 66X/
              *YLSL
                                                                                200500
              'YLCL
                         YUCL 1 + 66X/
                                                                                200600
              TYMAP
                         YUCL 1 . 66X/
                                                                                200700
              YUSCP
                         YSID + 66X/
                                                                                200800
              TYLSCP
                         YSID . 66X/
                                                                                200900
              *YUSLCP
                         YSSL 1 . 66X/
                                                                               201000
                         YSSL . . 66X/
              'YLSLCP
                                                                               201100
              *ZUSCP
                         ZUCL + 66X/
                                                                               201200
              'ZLSCP
                         ZLCL + , 66x/
                                                                               201300
              *ZUSLCP
                         ZUSL . . 66X/
                                                                               201400
              'ZLSLCP
                         ZLSL',66X/
                                                                               201500
              'ZSSL
                         ZSID++66X)
                                                                               201600
```

Table 8. Concluded

END	201700
	201800
BLOCKDATA	201900
COMMON /RA/X(10)+Z(10)+TH(10)+H(10)+R(10)+R(10)+B(10)+H(10)+	202000
1 CURVE(10).DIR(10).X1(10).X2(10).Z1(10).Z2(10).T1(10).T2(10).	202100
2 KSI(10) .ETA(10) .RHO(10) .XQ(11) .ZQ(11) .TQ(11.2) .KRV(3) .CL(2) .	202200
3 KV(2) • V(10.3.3) • SL(3) • RHOSL(10.3) • CRYSL(10.3)	202300
4.VB(10.3.2.8).CRVB (10.8).XMAPAX(10).ZMAPAX(10).XSID(10).ZSID(10).	202400
5 BLB(8)	202500
COMMON /IA/ICASE(10).JCASE(30).NCASE(7).NP(3).NS(3).NB(8)	202600
COMMON /RS/KI:MI:KJ:MJ:MIJ:LINE;BLNK:RINF;ERRMAX;OPDN;ELLX;	202700
1 YUCL-ZERO-FORE-AFTE-TOPS-BOTS-OPUP-OGIV-P10180-XJ-ŽJ-XI-ZI-M1-	202800
XI.BI.THI.BJ.AI.CA.CHIJ.S.THIJ.S.THIJ.SIJ.AI.HJ.BJ.A4.DX	202900
3.TYPE.ASYM.UPFO.UPAF.UPTO.UPBO.DNFO.DNAF.DNTO.DNBO	203000
COMMON /IS/IR.IW.IF.IP.IPRINT.NSEG.NSEGP1.ISTOP.JSTOP.KSTOP.	203100
1 I.J.L.IP1.LM1.II.NPTS.IZ.ISL.LB.N.NM1.I3	203200
COMMON /LA/SLOPE(10).IDEN(80).CARD(80)	203300
COMMON /LS/TI.T	203400
REAL K•M•KSI•KI•MI•KJ•MJ•MIJ•LINE•KV•KRV	203500
LOGICAL*1 SLOPE+IDEN+CAHD+TI+T	203600
DATA KRV/'LINE', 'OGIV', 'ELLX'/.LINE/'LINE'/,CL/'ZUCL', 'ZLCL'/.	203700
1 KV/'KV5 ','KV0 '/.SLOPE/10"'F'/.T/'T'/	203800
DATA RHO.X1,X2,Z1,Z2,T1,T2,K51,ETA/90+1,E70/	203900
DATA X-Z- R-H-K-H-B-TH/ 80-1-E70/	204000
DATA	204100
<pre>1 ERRMAX/1.E-5/.DIR/10**OPDN*/.OPDN*/.ELLX/*ELLX*/.</pre>	204200
2 YUCL/'YUCL'/',ZERO/0./'FORE/ 'FORE'/'AFTE/'AFTE'/'AFTE'/'TOPS'/'	204300
3 BOTS/'BOTS'/*OPDN/'OPDN'/*OGIV/'OGIV'/*OPUP/'OPUP'/	204400
DATA X9-29-T9/44*1-E70/	204500
DATA V/ 90+1.E70/-SL/'ZUSL''ZLSL''YSSL'/	204600
DATA RHOSL/30+1.E70/.CRVSL/30+'LINE'/	204700
DATA VB/480+1.E70/.CRVB /80+!LINE!/.TYPE/!SYMM!/.ASYM/!ASYM!/	204800
DATA BLB/'ZUCL'+'ZLCL'+'YSID'+'ZUSL'+'ZLSL'+'YSSL'+'ZSID'+'ZMAP'/	204900
DATA XMAPAX-ZMAPAX-XSID-ZSID/40+1.E70/	205000
DATA UPFO » UPAF » UPTO » UPBO » DNFO » DNAF » DNTO » DNBO	205100
1 /'UPFO': 'UPAF': 'UPTO': 'UPBO': 'DNFO': 'DNAF': 'DNTO': 'DNBO'/	205200
DATA JCASE/582.774.834.526.624.006.010.012.136.590.	205300
1 626,628,632,014,138,140,142,638,832,262,	205400
2	205500
-9*0/	205600
DATA NCASE/ 1. 2. 3. 4. 6.	205700
1 7, 8 /	205800
END	205900

### Table 9. Subroutine Description

Subroutine Name	Function
Main Program	Overall program control
INIT	Initializes certain program variables
INPT	Reads and prints input data
CASE	Determines the case number ICASE(I) of the input option for each segment (see Table 2)
SEGE	Determines the equation of each segment and computes certain intersection information
INTR	Determines intersection points of successive segments for which only segment equations are known to that point in code
QARY	Assembles coordinate and slope data in convenient arrays for building QUICK input deck (QUCK)
PRNT	Prints the QUICK input arrays assembled by QARY
R¢FL	Determines the equations of rounds and fillets and computes intersections with body line segments, prints results
QUCK	Builds the QUICK input deck for the symmetrical case with asymmetrical slices
LIST	Prints the QUICK input deck (if unit IF is not the card punch) for checking by user
SLCE	Performs geometry computation for slicing planes, prints results
RFSL	Determines the equations of rounds and fillets and computes intersections with slicing plane segments, prints results
СфРУ	Assembles coordinate and slope data in convenient arrays for building QUICK input deck for asymmetrical case (QK3D)

### Table 9. Concluded

Computes default values of certain body BLCK

line coordinates and checks validity of

previously computed coordinates for asymmetrical case

Builds the QUICK input deck for the asymmetrical case with slices OK3D

Initializes certain program constants BL¢CKDATA

Table 10. Variable Description

Variable Name	Type*	Description
В	RA	Z intercept of straight body line segment
H	RA	X coordinate of ogive center
I	IS	Intersection subscript
J	IS	Scratch integer
K	RA	Z coordinate of ogive center
L	IS	Scratch integer
M	RA	Slope of body line segment
N	IS	Scratch integer
R	RA	Radius of ogival segment
S	RS	Variable used in selection from multiple intersections, S=±1.
T	LS	= 'T'
V(I,J,L)	RA	Slice geometry array: J=1, X coordinate; J=2, Z (or Y) coordinate; J=3, slope angle (deg); L=1, upper slice; L=2, lower slice; L=3, side slice
x	RA	Input array for X coordinate of body lines
Z	RA	Input array for Z (or Y) coordinate of body lines
Ai	RS	Scratch variables i=1,2,
BI	RS	Z intercept of Ith segment
BJ	RS	Z intercept of I-1st segment
CL	RA	Body line names used in symmetrical case to assemble QUICK input deck

DX	RS	Round-off compensation variable
HI	RS	X coordinate of ogive center for Ith segment
НJ	RS	X coordinate of ogive center for I-1st segment
IF	IS	Unit number (data set reference number) of QUICK file written by QUCK and QK3D
IP	IS	Unit number of facility card punch
IR	IS	Unit number for input data
IW	IS	Unit number for facility printer
Ii	IS	Scratch integers, i=1,2,
KI	RS	Z coordinate of ogive enter for Ith segment
КЈ	RS	Z coordinate of ogive center for I-1st segment
KV	RA	Segment type (line or ogive)
LB	IS	Overall D¢ variable for asymmetrical case, LB=1, upper centerline; LB=2, lower centerline; LB=3, side (see Fig. 7)
MI	RS	Slope of Ith segment (straight line)
MJ	RS	Slope of I-1st segment (straight line)
NB (LB)	IA	Number of segments in the body line seg- ments in asymmetrical case
NP	IA	NS+1
NS(L)	IA	Number of segments in the three slicing planes, L defined as for V above
RI	RS	Radius of ogive for Ith segment
RJ	RS	Radius of ogive for I-1st segment

SL	RA	Body line names for slicing planes
тн	RA	Input array for local slopes of body lines at end of segment
TI	LS	=SL\perpoonup PE(I)
TQ(I,J)	RA	Local segment slope (degrees): J=1, at beginning of Ith segment; J=2 at end of Ith segment; input to subroutine QUCK
Ti	RA	Slope (degrees) of fillet or round: i=1, beginning of fillet; i=2, end of fillet
VB(I,J,N,L)	RA	Coordinates and slopes of the eight body lines of the asymmetrical case:     J=1, X coordinate     J=2, Z coordinate     J=3, slope (deg); N=1, beginning of         Ith segment, N=2 end of Ith segment         (N=2 used only with J=3)  L=1 Upper centerline (ZUCL) L=2 Lower centerline (ZUCL) L=3 Side (YSID) L=4 Upper slicing plane (ZUSL) L=5 Lower slicing plane (ZUSL) L=6 Side slicing plane (YSSL) L=7 Side (ZSID) L=8 Map axis (ZMAPAX)         (see Fig. 8)
xı	RS	X coordinate of end of Ith segment
хЈ	RS	X coordinate of end of I-1st segment
XQ	RA	<pre>X coordinate of beginning of Ith segment (I=1,NSEG) and end of last segment (I=NSEGP1), input to QUCK</pre>
Xi	RA	X coordinate at intersections of fillet or round with body lines, i as above in Ti
ZI	RS	Z coordinate at end of Ith segment
ZJ	RS	Z coordinate at end of I-1st segment

ZQ	RA	<pre>Z coordinate of beginning of Ith segment (I=1,NSEG) and end of last segment (I=NSEGP1), input to QUCK</pre>
Zi	RA	Z coordinate at intersections of fillet or round with body lines, i as above in Ti
BIJ	RS	Z intercept
BLB(L)	RA	Body line names for QK3D, L as above for VB
DIR	RA	Selection indicator variable for multiple root intersections
ETA	RA	Z coordinate of round or fillet center at end of Ith segment
IP1	IS	Scratch integer, =I+1
ISL	IS	Slice indicator: ISL=0 no slices ISL=1 slices present
KRV	RA	Segment type, line or ogive
KSI	RA	X coordinate of round or fillet center at end of Ith segment
LM1	IS	scratch integer, =L-1
MIJ	RS	Tangent of straight line slope angle
NM1	IS	Scratch integer, =N-1
RΗφ	RA '	Input variable for radius of round or fillet at end of Ith segment
THI	RS	Slope (deg) at end of Ith segment
ТНЈ	RS	Slope (deg) at end of I-1st segment
AFTE	RA	='AFTE', possible DIR value (aft solution)
ASYM	RA	='ASYM', possible TYPE value
BLNK	RS	= 'bbbb' (four blank characters)
ВфТЅ	RS	='B $\phi$ TS', possible DIR value (bottom solution)

CARD(80)	LA	Card image array
CRVB(I,L)	RA	Curve type indicator for QUICK, CRVB = 'LINE' or 'ELLX'
ELLX	RS	='ELLX'
FφRE	RS	='F $\phi$ RE', possible DIR value (fore solution)
IDEN(80)	LA	Input variable for identification data
LINE	RS	='LINE', possible CURVE value
NPTS	IS	Number of stations at which QUICK will print out geometry data if IPRINT≠0
NSEG	IS	Number of segments for symmetrical case
φGIV	RS	='\psi GIV', possible CURVE value
φPDN	RS	='φPDN', possible DIR value (open down)
φPUP	RS	='φPUP', possible DIR value, (open up)
RINF	RS	Real infinity, = 1.E69
THIJ	RS	Straight-line slope (deg)
TLSL } TSSL } TUSL }	RA	Input arrays for local slopes of the lower slice, side slice, and upper slice segments, respectively
XLSL XSSL XUSL	RA	Input arrays of X coordinate of end of lower, side, and upper slicing plane segments, respectively
ZLSL ) YSSL } ZUSL }	R <b>A</b>	Input arrays of Z coordinate of end of lower, side, and upper slicing plane segments, respectively
ΤφPS	RS	='Tops', possible DIR value (top solution)
TYPE	RS	<pre>Type body: TYPE ='SYMM', symmetrical</pre>
XSID	RA	X coordinate of side body line (Z)

YUCL	RS	'YUCL'
ZERφ	RS	Real zero = 0.0
ZSID	RA	Z coordinate of side body line
CRVSL(I,L)	RA	Curve type (='LINE' or 'ELLX') for slicing planes, L as in V above
CURVE	RA	Curve type for Ith segment, ='LINE' or '\phiGIV'
ICASE	IA	Integer for identifying input option (see Table 2)
INPUT		Namelist name for input data
ISTφP	IS	Error indicator to halt execution when set different from zero
JCASE	IA	Integer array used to determine ICASE (see Table 2)
JSTφP	IS	Error indicator similar to ISToP
КЅТФР	IS	Error indicator similar to ISTOP
NCASE	IA	<pre>ICASE values which input options may not follow ICASE = 5 or 9</pre>
RНфLS	RA	<pre>Input array for lower slice fillet/round radii</pre>
RH $\phi$ SL(I,L)	RA	Array of fillet/round radii, L as in V above
RHφSS	RA	<pre>Input array for side slice fillet/round radii</pre>
RHφUS	RA	<pre>Input array for upper slice fillet/round radii</pre>
SL¢PE	LA	Variable to indicate Ith segment is to be tangent to end of I-1st segment
ERRMAX	RS	Computational error limit, = 5.E-5

### Table 10. Concluded

IPRINT	IS	Print indicator: IPRINT≠0 causes KWIKN¢SE to generate cards to make QUICK exercise the completed math model
NSEGP1	IS	=NSEG+1
PI <b>ф180</b>	RS	π/180
XMAPAX	RA	X coordinate of map axis on input
ZMAPAX .	RA	Z coordinate of map axis on input

<sup>\*</sup>R=real, I=integer, L=logical, A=array, S=scalar